

Teaching Practice of Virtual Reality Development Course Based on Coin Collecting Game Project

Miao Yao*, Ju Li, Aizila Dawuxiati

School of Information Engineering, Urumqi Vocational University, Urumqi 830000, Xinjiang Uygur Autonomous Region, China

*Corresponding author: Miao Yao, 845308358@qq.com

Copyright: © 2024 Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY 4.0), permitting distribution and reproduction in any medium, provided the original work is cited.

Abstract: In response to traditional process-based project-based teaching, each chapter of the course is structured around an independent project. However, this approach lacks coherence in project content, leading to low participation and enthusiasm among students in the classroom. This article combines the characteristics of virtual reality development courses and introduces a novel teaching practice model driven by a gold coin-picking game project. The course content is seamlessly integrated into the practical aspects of the gold coin-picking game project. The course learning process mirrors the virtual reality game development process, effectively engaging students and fostering active learning, resulting in a sense of achievement.

Keywords: Virtual reality; Game project; Teaching practice

Online publication: February 25, 2024

1. Introduction

“The Outline of The 14th Five-Year Plan for Economic and Social Development and Long-Range Objectives Through the Year 2035 of the People’s Republic of China ”^[1], as well as the “Action Plan for the Integration and Development of Virtual Reality and Industry Applications (2022–2026)”^[2], both explicitly point out that virtual reality, including augmented reality and mixed reality, is an important frontier in the new generation of information technology, as well as a significant prospective field in the digital economy. The virtual reality technology sector plays a crucial role in the strategic layout of the national digital economy, with the development of virtual reality courses being a key priority. In the teaching practice of virtual reality development courses, it is crucial to consider students’ motivation for learning and the cultivation of their ability for independent innovation.

2. Issues in the teaching of virtual reality development courses

2.1. Weak coherence in course content

Virtual reality development courses emphasize practical teaching. However, certain textbooks organize course

content into individual projects based on chapters, resulting in fragmented content lacking coherence between projects. Consequently, students struggle to apply and utilize learned knowledge and skills in practice, leading to a lack of sustained enthusiasm for learning.

2.2. Lack of creativity in project design

Virtual reality projects inherently offer high interactivity and immersive experiences for learners and developers ^[3]. However, current project-based virtual reality development courses lack creativity and student-centered project design ^[4]. Instead, the focus remains on teaching individual chapters' knowledge and skills, predominantly led by instructors. This approach fails to effectively engage students and stimulate their enthusiasm and interest in learning. For instance, in the "Fundamentals of Unreal Engine (UE4)" ^[5], the chapter on the material system primarily covers the creation of metal and concrete materials, which fails to capture students' interest.

3. Teaching practice of virtual reality development course driven by coin collection game project

3.1. Course objectives

Project-driven teaching cultivates students' problem-solving and comprehensive practical abilities through continuous engagement ^[6]. This course prioritizes students' passion for gaming, centers around their needs, and emphasizes tangible outcomes. It begins by presenting the coin collection game project, which students create upon completing the virtual reality development course, thereby capturing their interest ^[7]. Subsequently, the coin collection game project is deconstructed into teaching components, including terrain system, material system, blueprint system, animation system, particle system, user interface (UI), and lighting system. Students acquire knowledge and skills through gameplay and project design, ensuring applicability to real-world scenarios and fostering continuous, long-term learning.

3.2. Introduction to coin collection game project

The coin collection game is a third-person perspective game in which players control the character using the "WASD" keys for movement and the spacebar for jumping. The game features a 60-second countdown and 20 preset coins randomly scattered across the scene. Each collected coin awards the player 10 points. To win the game, players must accumulate a total score of 100 within the allotted time. Failing to reach this score results in a loss.

3.3. Breakdown of coin collection game project

In this course, we use the coin collection virtual reality game project as an example and divide it into 7 sub-modules: scene terrain creation, material creation, blueprint creation, particle effects creation, character animation creation, game UI creation, and lighting effects creation.

3.3.1. Scene terrain creation module

Steps for designing the coin collection game scene terrain:

- (1) Determine the game theme: Establish the game's theme as adventure-based.
- (2) Design a background story: Create an engaging narrative for the game scene to immerse the player in the story.
- (3) Define the concept of the game scene: Based on the theme and background story, develop the concept

of the game scene, including level settings, map design, props, and characters.

- (4) Create the game map: Use design tools like Unity or Unreal Engine to construct the game map, considering terrain, buildings, and obstacles.
- (5) Define interactive elements: Determine interactive elements such as switches and mechanisms to add interest and challenge.
- (6) Design game levels: Develop levels with clear objectives and challenges to match the game's difficulty and player progression.
- (7) Test and optimize: Conduct testing to ensure smooth gameplay and adjust the scene based on feedback.
- (8) Beautify and optimize: Enhance the visual appeal of the scene by adding details and effects.

3.3.2. Material creation module

Steps for creating coin materials:

- (1) Gather reference materials: Before starting material creation, gather relevant references, including the appearance, color, and texture of the coins.
- (2) Create basic materials: Utilize image editing software like Photoshop or material editing software such as Unreal Engine to develop basic materials. Define the material's appearance using textures, colors, lighting, etc.
- (3) Create texture maps: Generate texture maps such as diffuse, normal, and roughness maps by capturing photos, painting, or using computer-generated methods. These maps define the details and surface characteristics of the material.
- (4) Determine material properties: Based on the coins' nature and their role in the scene, establish the material's physical properties, such as metalness, smoothness, and transparency, affecting its rendering in the game.
- (5) Create material channels: Apply texture maps and other properties to material channels, including the diffuse, normal map, and ambient occlusion channels. These settings dictate the material's appearance under different lighting conditions.
- (6) Debug and optimize: Test and optimize the materials by assessing their effects in the game engine. Adjust parameters related to lighting, color, and textures to achieve improved visual results.
- (7) Export and apply: Export the created materials in a game engine-supported format and apply them to objects in the game.

3.3.3. Blueprint creation module

Steps for creating the coin collection game blueprint:

- (1) Determine the core mechanics and functions of the game: Define the game's essential mechanics and functions, including objectives, gameplay, user interface, and character behavior.
- (2) Create a blueprint design document: Begin by drafting a design document to document the game's logic and functions.
- (3) Use the blueprint editor: Employ the blueprint editor of the game engine (e.g., Unreal Engine's blueprint editor) to develop the blueprint. This interface enables developers to create and connect logic and functions graphically.
- (4) Create blueprint classes and variables: Within the blueprint editor, establish blueprint classes and variables. These classes define object behavior and attributes, while variables store and pass data.

- (5) Add blueprint nodes and connections: Utilize blueprint nodes to represent various functionalities and logic. These nodes can trigger events, execute actions, and perform conditional checks. Connect them to define the game's flow and interaction.
- (6) Debug and test: Once the blueprint is complete, debug and test it to ensure its logic and functions operate as intended. Game engines typically offer debugging tools and testing environments for identifying and resolving errors.
- (7) Integrate into the game: Upon successful testing, integrate the blueprint into the game.

3.3.4. Particle effects creation module

Steps for creating the coin collection game particle effects:

- (1) Determine game requirements: Begin by identifying the desired particle effects for the coin collection game, such as using sparkles or beams to indicate coin collection.
- (2) Create particle system: Utilize the particle system tools of the game engine (e.g., Unreal Engine's Cascade) to generate particle effects. Customize particle shapes, colors, sizes, etc., and configure emitters and emission rates accordingly.
- (3) Design particle effects: Use emitter variations to depict coin appearances, then manipulate particle emission, rotation, scaling, fading, etc. to illustrate coin movement and disappearance.
- (4) Add particle textures: Enhance the realism of particle effects by incorporating textures to modify particle appearances. Utilize different textures to represent various coin shapes and colors.
- (5) Adjust particle parameters: Fine-tune particle parameters, including emitter position, speed, angle, particle lifespan, size, color, and response to forces and gravity, to achieve desired effects.
- (6) Testing and optimization: Conduct thorough testing and optimization to ensure smooth operation and alignment with game requirements.
- (7) Integration into the game: Integrate the particle system as a game object for corresponding coins and activate effect playback when the player collects the coins.

3.3.5. Character animation creation module

Steps for creating coin collection game character animation:

- (1) Character design: Start by designing and creating the characters for the game. Determine their appearance, shape, size, etc., as well as their basic actions such as standing, walking, jumping, and coin collection.
- (2) Animation planning: Before beginning character animation creation, establish an animation plan. Determine the necessary actions and transition animations for the character, along with triggers for these animations (e.g., player input or specific event triggers).
- (3) Animation software: Choose appropriate animation software, such as Animator in Unity or Spine software.
- (4) Animation skeleton: Based on the character's design, create a basic skeleton or skeletal system. This skeleton serves as the foundation for character animation and allows control and manipulation of various parts of the character.
- (5) Keyframe animation: Use the keyframe function in the animation software to create animations for the character's actions, including basic actions like standing and walking, as well as specific actions like jumping and coin collection.
- (6) Transition animations: Develop transition animations to smoothly switch between different actions,

such as transitions from standing to walking or from walking to jumping. Utilize the transition function in the software to achieve these effects.

- (7) Animation optimization and adjustment: Optimize and adjust the character animations to ensure smoothness and alignment with game requirements.
- (8) Integration into the game: Once satisfied with the character animations, integrate them into the game.

3.3.6. Game UI design module

Steps for designing coin collection game UI:

- (1) Design interface: Determine the overall style and theme of the game UI, considering colors, images, fonts, and layout for consistency and attractiveness.
- (2) Interface elements: Define necessary elements like title bars, buttons, text boxes, and icons, designing their appearance based on game requirements.
- (3) Layout design: Develop a logical layout considering functionality and content, ensuring elements are positioned well for user interaction.
- (4) Image assets: Create required assets such as backgrounds and icons according to the interface design.
- (5) Text and font: Select appropriate fonts and styles for readability and ease of use.
- (6) Animation and transition effects: Enhance interface appeal with animations like button clicks and page transitions.
- (7) User interaction: Ensure elements are interactive and navigable.
- (8) Integration into the game: Once the UI design is finalized, integrate it into the game.

3.3.7. Lighting effect creation module

- (1) Steps for creating coin collection game lighting effect:
- (2) Set light source position: Determine the optimal position for the light source in the game scene to achieve the desired effects.
- (3) Determine light source type: Select the appropriate type of light source – point lights, spotlights, or directional lights – based on game requirements.
- (4) Set light source properties: Adjust properties such as intensity, radius, and color to fine-tune the lighting effect.
- (5) Render the scene: Utilize the rendering functionality of the game engine or graphics library to apply the lighting effect to objects in the scene.
- (6) Adjust lighting effect: Experiment with different light source positions, properties, and scene layouts to achieve desired lighting aesthetics.
- (7) Lighting special effects: Enhance the lighting effects with additional effects such as shadows, halos, and glare to increase virtual realism.
- (8) Testing and debugging: Evaluate the performance of the lighting effect in the game and make necessary adjustments for optimal results.

4. Conclusion

By analyzing the issues in teaching virtual reality development courses and embracing students' interests, a new practical teaching model for these courses is proposed, centered around the virtual reality coin collection game project. This model integrates the development and design process of the coin collection game into the course curriculum, thereby enhancing students' engagement and fostering their engineering project thinking. This

integration facilitates the mutual enhancement of knowledge transmission and value dissemination.

Through the implementation of teaching reforms in virtual reality development courses, students' enthusiasm for learning has been revitalized, fostering an environment of independent learning and active exploration. Consequently, this has significantly enhanced the teaching quality and students' academic achievements.

Disclosure statement

The authors declare no conflict of interest.

References

- [1] National Development and Reform Commission (NDRC) People's Republic of China, 2022, The Outline of The 14th Five-Year Plan for Economic and Social Development and Long-Range Objectives Through the Year 2035 of The People's Republic of China, Beijing.
- [2] The People's Republic of China's Ministry of Industry and Information Technology, Ministry of Education, National Cultural and Tourism Administration, et al., 2022, Action Plan for the Integrated Development of Virtual Reality and Industry Applications (2022–2026), Beijing.
- [3] Tan M, Hua H, Xiong Q, et al., 2023, Multistrength Graded Virtual Tactile Feedback Based on Microcurrent Stimulation. *Journal of Computer Research and Development*, 60(9): 2015–2027.
- [4] Hu Z, Wang Y, Feng W, et al., 2023, Exploration and Practice of “Student-Centered” Engineering Hybrid Teaching Course Design. *Higher Education Teaching Journal*, 9(2): 158–161.
- [5] Yao L, 2018, Unreal Engine (UE4) Technical Foundation. Electronic Industry Press, Beijing.
- [6] Yang S, Zhou X, 2023, Reform and Reflection on Program Design Course Based on Project-Driven Teaching. *Science and Education Forum*, 2023(9): 87–91.
- [7] Cui Y, Zhang Y, Qiu Z, et al., 2023, Design and Practice of an Interest-Centered Teaching Innovation Model in Organizational Biology and Embryology Classroom. *Journal of Anatomy*, 46(2): 164–167.

Publisher's note

Bio-Byword Scientific Publishing remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.