

Technical Report on the Implementation of a 3D character motion engine

Module Code: CPT205

Name : Xu Chen

ID: 2257453

Degree Programme: Information and Computer Science

1 Introduction

Traditional 2D games, while dominant in the early history of gaming, had inherent limitations. The most significant of these was the lack of depth and spatial freedom, as players could only interact within a flat plane, which restricted the richness and immersion of the gaming experience. With the advancement of computer graphics technology, 3D games gradually became the direction for future development, particularly in racing and flight simulation games, where the introduction of 3D provided players with a more realistic and dynamic environment. However, despite the new visual experiences offered by 3D technology, these games still faced challenges, particularly in the design of camera perspectives.

Traditional 3D racing and flight games often employed fixed or simplistic camera angles, which could easily cause players to lose their sense of direction during high-speed movement, or make the game's visual presentation appear stiff and unnatural. Especially in fast-changing environments, the lack of flexible camera control prevented the full potential of 3D space from being realized, thus posing a key issue in 3D game development: how to solve these problems through innovative camera design.

*Both **Super Mario 64** and **The Legend of Zelda: Ocarina of Time** not only broke through the technical limitations of their time but also provided important lessons for the development of 3D games, particularly in camera systems. Super Mario 64, with its groundbreaking 3D world and free camera control, offered players an unprecedented sense of immersion, and its design principles continue to influence modern open-world games. The Legend of Zelda: Ocarina of Time took it a step further with more complex environmental interactions and intelligent camera designs, further enhancing the visual presentation and player control within the game.*



Figure1 : Screenshots of Super Mario 64 and The Legend of Zelda: Ocarina of Time

In this report, I conduct an in-depth study of the technical details of these two games, focusing particularly on how they utilize camera systems to enhance the player experience. By analyzing the camera control techniques, dynamic camera transitions, and player-environment interaction mechanisms within the games, I explore how to achieve smoother and more natural interaction experiences, aiming to create an immersive 3D world with a sense of time.

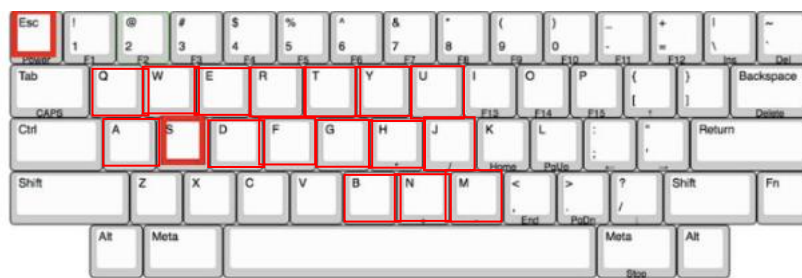
2 Readme

2.1 Mouse Movement

Horizontal movement (left-right): Controls the left-right rotation of the camera.

Vertical movement (up-down): Controls the up-down rotation of the camera.

2.2 Keyboard Controls



For human control :

W: Move the character forward, while moving the camera position to follow the character.

A: Move the character to the left, while moving the camera position to follow the character.

S: Move the character backward, while moving the camera position to follow the character.

D: Move the character to the right, while moving the camera position to follow the character.

Q: Rotate the character clockwise, making the character rotate. The camera perspective will adjust accordingly, keeping the character at the center.

E: Rotate the character counterclockwise, making the character face left. The camera perspective will adjust accordingly, keeping the character at the center.

F: Increase the character's movement speed, allowing the character to move faster.

R: Decrease the character's movement speed, making the character move slower.

For camera control :

Y: Increase the camera's vertical offset, moving the view upward.

H: Decrease the camera's vertical offset, moving the view downward.

T: Increase the camera's horizontal offset, moving the view to the right.

U: Decrease the camera's horizontal offset, moving the view to the left.

J: Increase the camera's forward and backward offset, moving the view forward.

G: Decrease the camera's forward and backward offset, moving the view backward.

N: Increase the camera's observation distance, moving away from the current view position.

M: Decrease the camera's observation distance, moving closer to the current view position.

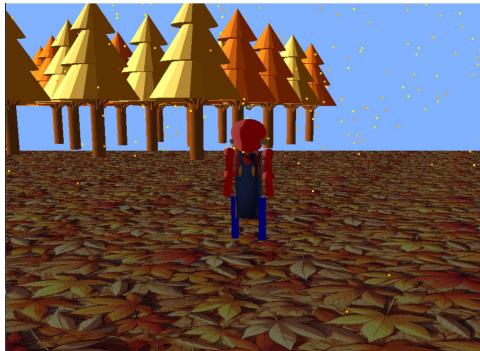
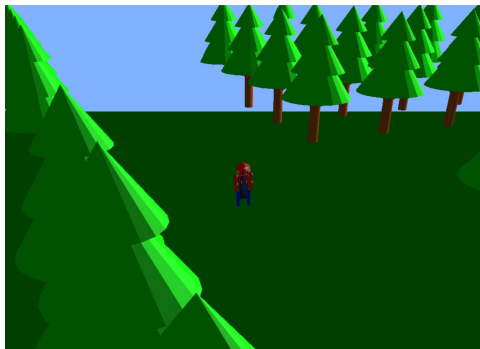
For time Accelerating :

B: Accelerate the speed of the light source, enabling a faster transition through the four seasons.

Exit :

Esc: Pressing the Esc key will exit the program. The program will terminate immediately.

3 Screen Shots



4 Design Overview

Character Rotation and Camera Perspective: Solved the issue of rotating the character with the Q and E keys while simultaneously adjusting the camera's view. Additionally, forward movement now follows the character's facing direction, rather than its original direction.

Arm Movement: The character's arms are designed to swing, with the rotation centered around the origin of the arm's circle.

Camera Follows the Character: The camera no longer moves freely, but instead follows the character's movement, ensuring a more natural and immersive experience.

Lighting and Material Properties: Introduced diffuse lighting and defined material properties, which enhances the visual quality of the rendered objects by adding a more realistic texture.

Dynamic Lighting and Seasonal Changes: Addressed the issue of controlling the light source's position and speed, and implemented a system to simulate the changing of seasons. The lighting speed accelerates to transition quickly through seasons.

Tree Modeling: Designed trees with branches that are inclined at 45 degrees, and applied random colors to the leaves. No leaves are drawn, but the branches are visible to represent seasonal changes.

Particle Effects for Snowfall and Falling Leaves: Implemented random movement for particles such as falling snow and leaves, contributing to a dynamic seasonal environment.

Seasonal Transitions and Color Changes:

Enabled the simulation of seasonal transitions and the ability to accelerate the viewing of these changes. The environment's colors shift accordingly with each season.

Texture Reflection and Diffuse Lighting: Addressed issues with character textures and reflection by applying diffuse lighting and specular highlights, improving the realism of reflective surfaces and the character's appearance.

5 Conclusion

In this project, I have successfully implemented a series of key features that enhance the realism and interactivity of a 3D game environment. By integrating dynamic camera control, character movement, and lighting systems, I addressed several

challenges inherent in 3D game design. The character now rotates seamlessly with camera adjustments, and its movement follows a natural path aligned with the character's facing direction. The arm swing effect adds a layer of realism, while the camera follows the character's movements, offering an immersive experience.

Furthermore, by introducing lighting techniques such as diffuse lighting and material properties, I achieved more realistic rendering of objects. The system for simulating seasonal changes, combined with the timer-driven dynamic lighting, provides a smooth and visually engaging transition through different times of the year. Particle effects such as snow and falling leaves further enhance the environment's dynamic feel, while tree models with inclined branches and random coloring contribute to the visual diversity across seasons.

The use of textures, along with improved reflection and diffuse lighting techniques, solves earlier issues and elevates the overall appearance of both characters and environments. These features combine to create a more interactive, immersive, and visually compelling 3D world, setting a solid foundation for further development and expansion in future projects.