Super Mario's AIUB Quest

American International University Bangladesh, AIUB

Course: CSC4118: Computer Graphics

Instructor Name:

DIPTA JUSTIN GOMES

Due Date (December 22, 2024)

Team Member List:

Name	ID	Contribution
Nur Rahatul Islam	23-51007-1	Introduction, Conclusion
Urmi Karmakar	22-47163-1	Description
Rakesh Karmaker	22-46862-1	Feature Set

Title: Super Mario's AIUB Quest

Introduction: Video games have always been loved by many, with classics like Super Mario being especially popular. Inspired by the success and charm of these timeless games, this project developed as part of a computer graphics course, will apply concepts and techniques learned throughout the semester to create a visually appealing and interactive 2D platformer. The game will feature smooth player mechanics, such as running, jumping, and interacting with the environment. In addition to basic movement, the gameplay will involve challenges such as coin collection, enemy avoidance, and interaction with different environmental elements. The aim of this project is to merge nostalgia for classic platformers with modern technical knowledge and creativity. By developing this game, the project will showcase the ability to design and create a fully functional and enjoyable 2D platformer. The game will serve as a demonstration of skills in game development, computer graphics, and problem-solving, while delivering an engaging and memorable experience for players.

Description of the environment and list of methods included in the project: The development of Super Mario's AIUB Quest computer graphics project utilizes the OpenGL library in C++, supported by Code Blocks IDE and GLUT for window management, input handling, and rendering. This project focuses on creating an interactive 2D platformer game with a visually appealing AIUB-themed environment. Graphics are rendered using OpenGL, employing texture mapping and real-time rendering techniques to produce detailed and immersive levels inspired by AIUB's campus. The game includes smooth player mechanics, such as running, jumping, and interacting with the environment, with collision detection ensuring seamless interactions between the character, obstacles, and collectibles. Keyboard inputs are processed using GLUT to provide intuitive and responsive control of the character's actions. Additionally, the AIUB-themed background enhances the visual appeal and creates a unique connection to

the setting, while background music and sound effects enrich the overall experience. By combining core computer graphics concepts with creative design, this project highlights the developers' ability to apply technical skills effectively to deliver a polished and enjoyable gaming experience.

Feature set: Feature Set with tasks broken down into components:

Player Features:

1. Moving left when pressing the left arrow key.
2. Moving right when pressing the right arrow key.
3. Jumping when pressing the spacebar.
4. Stopping the move when no key is pressed.
5. Displaying the player spirit.
6. Preventing the player from falling through the ground.
7. Making the player respawn at the starting point after falling.
8. Displaying a "Game Over" message when the player loses.
9. Adding a score counter that increases with each coin collected.
10. Resetting the score when the level is restarted.
11. Adding platforms the player can walk on.

Game Object Features:

12. Adding blocks that the player can jump on.
13. Adding coins floating above the platforms.
14. Adding enemies that stay in one place.
15. Adding spikes that cause the player to lose.
16. Adding a flagpole at the end of the level.
17. Adding decorative objects like trees.

18. Making blocks that disappear when the player jumps on them.
19. Adding checkpoints where the player respawns.
20. Adding walls that stop the player from leaving the level.
21. Making AIUB campus buildings.

Environment Features:

22. Making AIUB campus roads
23. Making AIUB campus Tress
24. Making other objects of the AIUB campus (like placing simple decorations)
25. Making a background that has AIUB campus
26. Adding the background
27. Making the ground green to represent grass.
28. Adding a pit that the player can fall into.
29. Adding clouds in the background.
30. Adding a basic bridge object for the player to cross.
31. Making an uneven ground with small hills.
32. Adding a sign that say, “Jump Here!”
33. Adding a water feature for decoration.
34. A background music during the level.

Audio and Visual Features:

35. Playing a sound effect when the player jumps.
36. Playing a sound effect when the player collects a coin.
37. Displaying the score at the top of the screen.
38. Showing a “Level Complete” message when the player reaches the flag.
39. Displaying health using small hearts.

40. Showing a “Pause” button on the screen.
41. Effects when the player touches spikes.
42. Adding a decorative “AIUB” sign in the background.
43. Adding simple particle effects when a coin is collected.
44. Creating a start button on the main menu.

Interactive Gameplay Features:

45. Creating an exit button on the main menu.
46. Adding a timer that counts down during the level.
47. Display a “Times Up” message when the timer runs out.
48. Adding a door that lets entry to the next level.
49. Making blocks that give a coin when hit.
50. Adding a platform that moves up and down.
51. Adding a static enemy that doesn’t move.
52. Adding an enemy that moves left and right.
53. Creating a button to restart.
54. Making a checkpoint system that saves progress.
55. Adding power-up that increases the player’s speed.
56. Display a pop-up message at the start of the level.
57. Add a block that disappears when touched.
58. Add multiple finish flag for a different level.

Level Features:

59. Create a basic level with platforms and coins.
60. Add more coins to collect in the level.

61. Add a small pit in the level.
62. Add a single moving platform.
63. Add a decorative lamp post.
64. Add a checkpoint in the middle of the level.
65. Include a decorative flag at the end of the level.
66. Add a “Start Game” button in the menu.
67. Display a “You Win!” message when the level ends.

Conclusion: This project aims to develop a 2D platformer game inspired by Super Mario, using OpenGL for graphics rendering. This project merges the nostalgic appeal of classic platformers with technical skills in computer graphics, applying key concepts learned during the course in a practical setting. In future versions of the game, several enhancements could be made. These include adding more complex levels with varied environments, improving character animations, and incorporating interactive objects. Advanced graphics techniques such as lighting effects, particle systems, and 3D modeling could enhance the visual quality of the game. Additionally, features like a scoring system, power-ups, and multiplayer modes could increase the depth and replay ability of the game. However, the project does have limitations. The quality of graphics will be constrained by the capabilities of OpenGL and the developers’ current skill levels, which may limit the complexity of some visual effects. Due to time and resource limitations, the scope of the project will focus on creating a polished prototype rather than a fully developed product. Performance optimization and compatibility testing across different systems may also pose challenges.