**Objective**

The core aim of this project is to craft an immersive 3D maze exploration game, utilizing advanced raycasting techniques. The vision is to create an experience that transcends conventional maze games by incorporating dynamic movement, spatial orientation, intricate visuals, and diverse interactive elements within a labyrinthine environment.

**Team Members**

1. Evans Mugambi - Developer of the Project

**Project Structure and Tasks**

**1. Walls!**

- *Window Setup*: Establish a functional window using SDL2, the foundation for rendering the game.

- *Raycasting*: Implement raycasting to dynamically draw walls within the game window, creating the illusion of depth and structure.

- *Camera Angle Modification*: Enable adjustment of the camera angle for testing purposes, enhancing flexibility in assessing the visual integrity of the maze.

- *Distinct Wall Color*: Define and apply varying colors for walls and the ground/ceiling, fostering visual differentiation.

**2. Orientation**

- *Visual Clarity*: Render walls facing NORTH and SOUTH distinct from those facing EAST and WEST, enhancing spatial orientation and visual cues for players.

**3. Rotation**

- *User Control*: Empower players to rotate the camera during gameplay, allowing exploration and interaction with the maze from different perspectives.

- *Input Modalities*: Implement multiple input options such as keyboard arrow keys or mouse movements for camera rotation.

**4. Move**

- Dynamic Movement: Enable fluid camera movement using keyboard inputs (e.g., w, a, s, d keys) for a seamless exploration experience.

- Collision Handling: Implement robust collision detection to prevent player camera from passing through walls, enhancing realism and challenge.

**5. Parser**

- *Map Extraction*: Develop a parser to extract maze layout information from a designated file, establishing a standardized format for storing and loading maze structures.

**6. Draw the Map**

- *Map Visualizatio*n: Display the parsed map on the game window, providing players with a visual representation of the maze.

- *Toggle Functionality*: Implement a toggle feature to enable or disable the map visualization during gameplay for enhanced immersion.

- *Line of Sight Integratio*n: Incorporate the player's line of sight within the map representation, enhancing strategic navigation.

**7. Coding Style & Documentation**

- *Holberton Standards*: Ensure strict adherence to the Holberton School coding style for consistency and readability.

- C*omprehensive Documentation*: Thoroughly document the codebase, following the Holberton School documentation format, to facilitate understanding and future development.

**8. Textures**

- *Visual Enhancement*: Introduce textures onto the walls, elevating visual aesthetics and enriching the immersive experience.

**9. Multi-task**

- *Simultaneous Actions*: Implement handling for multiple inputs, ensuring smooth and intuitive gameplay even when multiple movement or rotation keys are pressed simultaneously.

**10. Ground Textures**

- *Environment Detailing*: Apply textures on the ground and/or ceiling to add depth and ambiance to the maze environment.

**11. Weapons**

- *Interactive Elements*: Incorporate textured weapons into the game, fostering interactive elements and potential gameplay enhancements.

**12. Enemies**

- *Adversarial Presence*: Introduce antagonistic entities within the maze environment, challenging players and adding strategic depth to the game.

**13. Weather Effects**

- *Atmospheric Elements*: Implement a dynamic rain feature that can be controlled via a key input, enhancing the ambiance and realism of the game environment.

**14. Extra Option**

- *Innovative Features*: Experiment and implement additional features such as shadows, unique lighting effects, or any innovative elements to elevate the gameplay experience.

**Implementation Plan**

*1. Phase 1: Foundational Development*

- Set up the development environment, integrating SDL2.

- Implement fundamental raycasting mechanisms for rendering walls.

*2. Phase 2: Gameplay Mechanics*

- Develop movement, rotation, collision detection, and map parsing functionalities.

- Create basic map rendering and initial wall texturing.

*3. Phase 3: Advanced Features*

- Introduce additional elements like enemy AI, weather effects, interactive objects (weapons), and ground/ceiling textures.

*4. Phase 4: Polish & Refinement*

- Ensure code compliance with Holberton coding standards and comprehensive documentation.

- Fine-tune graphics, optimize performance, and incorporate any extra features or innovations.

**Conclusion**

MazeMaster 3D is envisioned as a captivating and intricate maze exploration adventure. The project aims to deliver an immersive gaming experience, leveraging raycasting techniques and a multitude of interactive elements to provide players with an engaging and visually stimulating adventure through labyrinthine landscapes.

*Sincerely,*

*Evans Mugambi*