A design documentation – Assignment 03 (Shuxun Zhou)

1. Original system analysis：
2. Overall architecture(for quoridor):

(1).QuoridorRenderer.java (game renderer)

(2).CellValue.java (Cell state enumeration)

(3).Quoridor.java (main game logic):

- Move Validation: Checks for wall obstruction and valid moves

- Jump Mechanics: Handles player interactions

- Wall Placement: Validates wall placement and prevents complete path blocking

- Path Finding: Uses a BFS algorithm to ensure game play is complete

(4).QuoridorPlayer.java (specialized player class):

- Extends `Player` to add Quoridor-specific functionality

- Manages the wall inventory (10 walls per player)

- Tracks position and target row

- Implements victory condition checks

1. Evaluation of the initial design architecture:

Game: The abstract class provides a clear template method implementation.

Play: Define a unified game loop interface.

Renderer: This interface allows each game to have unique display logic.

Board: Handle variable-size game boards and support dynamic aspect ratio settings.

1. Later adoption:

Minimal Expansion

- Advantage: Maintains existing architecture with minimal changes

- Disadvantage: May require compromising certain design ideals

1. Contribution:

- Design and implement the main game (Move players, build walls, detect victory) `Quoridor.java`

- Modify the game player class content of "QuoridorPlayer.java"