Quarto

Software Requirements Document
CSC 322 Software Engineering

Quarto! is a two-player board game by GiGaMic. It is like a 4x4 version of tic-tac-toe that a player wins by getting four in a row. There are 16 different game tokens, each with a unique combination of four binary characteristics. This project creates a video game version that allows two people to play against each other in two-player mode, or for a person to play against a computer opponent in single-player mode. The game can be played remotely on two computers that are communicating through a Quarto server, or on a single computer if both players take turns sharing the same mouse and monitor.

Feature #1: Networked Game Engine

Provide a network-based game engine that allows players on different computers to play against each other over the internet (or on the same computer).

Functional Requirements:

- A Quarto server program shall host each game and be accessible to clients through the TCP/IP network stack (i.e., RFC 1122).
- The Quarto server shall be responsible for enforcing the rules of the game.
- Before each turn, the server shall provide each client with the current state of the game.
- After each turn, the server shall validate the moves made by clients.
- The Quarto Client shall display an accurate view of the gameboard at each phase of the game and accept player input in response.
- The Quarto Client shall communicate each player's move to the server.
- The Quarto Client shall allow players to claim victory and then wait for verification of the victory from the Quarto Server.
- Quarto Clients shall acknowledge and display game-ending messages delivered by the Quarto Server.

Non-Functional Requirement:

- Client programs shall be able to play a game on any accessible Quarto Server. The Quarto Server could be internet-based, intranet-based, or running on the same computer as the client.
- The Server shall listen for incoming connections on a TCP port that is set at runtime.
- The communication protocol between the Server and clients shall be ASCII based.
- The software package shall include both the server software and the client software so that all users can play the game locally without connection to an internet server.

Acceptance Criteria:

- Two human players must complete a Quarto game while sharing the same computer.
- Two human players must complete a Quarto game while running the client software on two different computers that are each able to reach the same server instance.
- A malicious client must attempt invalid moves and the server must detect the errors and declare the malicious client's opponent as the winner.
- A single human player must complete a Quarto game against a computer opponent.
- A client must claim an invalid victory and the server respond within the game rules.
- A client must claim a valid victory and the server respond to both clients by ending the game and recording the winner.

Feature #2: Improved Graphics

Create a 2D/2.5D game board display that allows players to customize the in-game graphics.

Functional Requirements:

- Players shall be able to select a legacy graphics mode that displays the original 2D shape-based game tokens.
- Players shall be able to select a 2.5D graphics mode that displays 3D game tokens on top of a 2D game board.

Non-Functional Requirement:

- The game board shall retain the same shape and size regardless of which game tokens are selected.
- Players shall be able to change the colors of their 2D legacy tokens.
- Players shall be able to choose between multiple themes of the newer 3D tokens.
- Each of the three single-player computer opponent (easy, medium, hard) shall have its own set of tokens

Acceptance Criteria:

- A human player must change the colors of the 2D tokens from one game to the next through a set of configuration widgets.
- A human player must switch from 2D tokens to 3D tokens from one game to the next through a set of configuration widgets.
- A human player must switch between 3D tokens from one game to the next through a set of configuration widgets.
- A human player who has played multiple games against each single-player opponent must correctly identify the opponent based on the token shape and color at least 75% of the time.

Feature #3: Create Bot Interface

Create an interface that allows players to build bots that can play the game autonomously.

Functional Requirements:

- At the beginning of each turn, bots shall receive the current state of the game from the client program.
- Each bot shall communicate its desired move to the client, which will pass the command to the server.
- The bots shall interface with the client software in such a way that the bot builder does not need to include any network programming or knowledge.
- The Quarto software shall prevent bots from delaying games with slow code.
- The Quarto software shall prevent cheating bots from taking invalid moves or changing the state of the game.

Non-Functional Requirement:

- Before each game, human players shall be able to configure a maximum turn length. If a human takes longer than the turn, the player loses.
- Bots shall always take their turn in one second of less.
- The client shall be able to distinguish between human players and bot players to enforce timing rules.
- Once a game has started, bots shall interface with the Quarto software through a single function call.
- Bots that throw exceptions or otherwise crash shall automatically lose without interrupting the client or server portions of the Quarto software.

Acceptance Criteria:

- Bots running on a local server must respond to each individual move within 1 second (by running against a local server, the criteria avoids the impact of network delays).
- A bot with an internal infinite loop must play 10 games in a row without crashing any software and the server must stop the game within 3 seconds.
- A bot that throws an exception must play 10 games in a row without crashing any software and the server must stop the game within 3 seconds.
- A bot must play against another bot 10 games in a row without crashing any software.
- A bot must defeat a human opponent 3 games in a row and the server must correctly identify the winner of each game.
- A bot must lose to a human opponent 3 games in a row and the server must correctly identify the winner of each game.
- A bot must play 10 games in a row with a unique invalid move and the server must declare the bot as the loser.

Feature #4: Single Player Mode

Create a single player mode that allows human players to compete against computer players of varying skill.

Functional Requirements:

- There shall be at least one "easy" computer player that makes all decisions without considering the state of the game board.
- There shall be at least one "medium" computer player that avoids giving its opponent a winning token. However, this computer player shall *not* use information about the state of the game board when choosing where to place its own tokens on the board.
- There shall be at least one "hard" computer player that avoids giving its opponent a winning token and, if given a winning token, will place the token in the winning position on the game board.
- All computer players shall claim victory if they place a winning token on the board.

Non-Functional Requirement:

- The computer player shall meet all of the requirements of the bots.

Acceptance Criteria:

- The computer player must meet all of the criteria of the bots.
- The easy computer player must be given a game winning move and *not* take the move 3 out of 5 times.
- The medium computer player must be put in a situation where the player is about to win and *not* give the player the winning token 3 out of 5 times.
- The hard computer must complete all the tasks of the medium computer and it must take the winning move when handed a winning piece 5 out of 5 times.