CSCI 3010 SP ‘23
HW4 - Design Document
Leif Anders

ChutesAndLaddersGame

This class would represent the game itself and would contain the logic for the game, such as moving the players and handling chutes and ladders. It would also manage the game board and the players' positions.

public:

void startNewGame()	Starts a new game of Chutes and Ladders. Prompts for Number of players and their names.
int rollDice()	Rolls the current players dice
void moveCurrentPlayer(int numSpaces)	Move the current players pawn across the game board with the given value
bool isGameOver()	Check end game condition

signals:

void playerTurnChanged(int playerIndex)	Signals when player turn has changed
void diceRolled(int value)	Signals when dice has been rolled
void playerMoved(int playerIndex, int old_pos, int new_pos)	Signals when the player has moved from old position to new position
void gameOver(int winningIndex)	Signals when player has reached goal index

private:

QVector<Player*> players_	Vector containing pointers to all the players of the game
GameBoard * gameboard_	Pointer to the game board object
int curr_PlayerIndex	Index referring to which player is currently taking their turn
int dice_value	Value of die roll

Player

This class would represent a player in the game and would contain information about the player's name and current position on the board.

public:

void setName(const Qstring &name)	Set the players name
QString getName() const	Get the players name
void setPosition(int position)	Set the players position
int getPosition() const	Get the players position

signals:

void positionChanged(int position)	Signals when players position is changed

private:

QString name_	Players name
int position_	Players position

GameBoard

This class would represent the game board and would contain information about the locations of the chutes and ladders, as well as any other special spaces on the board.

public:

void setBoardSize(int size)	Set the size of the board
int getBoardSize()	Gets the size of the board
void setChute(int start, int end)	Set the start and end position of the chute
void setLadder(int start, int end)	Set the start and end position of the ladders
int getNextPosition()	Computes the next location of player based on roll, accounting for chutes and ladders

signals:

void chuteEncountered(int start, int end)	Signals when player lands on start of chute
void ladderEncounterd(int start, int end)	Signals when player lands on start of ladder
void winEncounterd(int end)	Signals when player lands on final position

private:

int boardSize_	Size of the board
Qmap<int, int> chutes_	The keys in this map represent the start positions of the chutes on the game board, and the values represent the end positions of the chutes.
QMap<int, int> ladders_	The keys in this map represent the start positions of the ladders on the game board, and the values represent the end positions of the chutes.

Chute

This class would represent a chute on the game board and would contain information about the starting and ending positions of the chute.

public:

int getStartPostion() const	Get the start position of the chute
int getEndPosition() const	Get the end position of the chute

private:

int startPosition	The start position of the chute
int endPosition	The end position of the chute

Ladder

This class would represent a ladder on the game board and would contain information about the starting and ending positions of the ladder.

public:

int getStartPostion() const	Get the start position of the ladder
int getEndPosition() const	Get the end position of the ladder

private:

int startPosition	The start position of the ladder
int endPosition	The end position of the ladder

MainWindow

This class would represent the main window of the game and would handle user input, display the game board and player information, and interact with the ChutesAndLaddersGame class to update the game state.

public:	
MainWindow()	Constructor
~MainWindow()	Deconstructor
slots:	
void newGame()	Connected to NewGame button
void quitGame()	Connected to QuitGame button
private:	
void setupUI()	Helper that creates and lays out the user interface elements in the Mainwindow
QWidget *centralWidget	Central widget where main content of window will be
QLabel *statusLabel	Displays information for user
QPushButton *newGameButton	New game button
QPushButton *quitGameButton	Quit game button