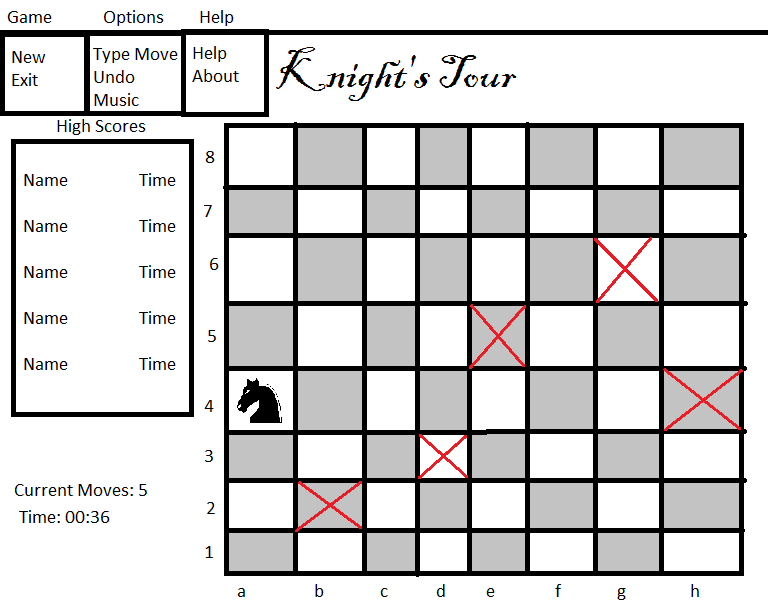
Knight’s Tour Initial Design – Terry and Kirill

Screen Layout:



The central feature of the game screen is the board, which shows the knight, its current position, and squares it has already visited.

The bottom-left corner includes a move counter (63 legal moves are required to win) and a timer (to keep track of scores)

The high score table will keep track of the 5 players having completed the tour with the lowest times.

Menu layout:

The ‘Game’ menu will have options to start a new game and to exit the program.

The ‘Options’ menu will have an option to move the knight by typing in a square position instead of clicking on the board, an option to undo the last move, a submenu allowing the player to change the background music, and a submenu to change the style of the knight and visited square markers (not shown)

The ‘Help’ menu will include options for showing instructions and an ‘about’ window.

Variables:

**boolean**[][] board = **new** **boolean**[8][8];

The array representing the board with board[0][0] being the bottom-left corner (aka a1). A value of **true** indicates that the knight may still visit the square and **false** means it cannot. This will be a global variable as the state of the board is important to several methods.

**int**[] knight = **new int**[2];

This array keeps track of the square the knight is currently on in the form [x, y]. This will be a global variable as the position of the knight is important to several methods.

**int**[][] undoMoves = **new int**[n][2];

This array will store past positions of the knight. n positions will be stored, where n is the number of undos given to the user. This will be a global variable as the list of undos should be persistent throughout the game.

**int** moveCount;

Keeps track of the number of moves made by the user.

**int** timer;

Keeps track of the time taken by the player to complete the tour.

Methods

**static** **int** moveKnight(**int** boardX, **int** boardY)

This method will move the knight to a different square on the board. It will first check if the move is legal. This will return 1 if the move is legal, so that moveCounter can be incremented. Otherwise, it will return 0.

**static int** undo()

This method will allow the user to take back a move by returning the knight to its previous position. It will then return -1 allowing moveCounter to be decremented. If the user has no more undos left, it will return 0.

**public boolean** mouseDown(Event evt, **int** x, **int** y)

Reads mouse clicks from the user, and calculates which square on the board was being clicked. Calls moveKnight() with the specified square.

**public void** newGame()

Starts a new game

**public void** typeMove()

Allows the user to type in a square co-ordinate (eg. f3), then attempts to move the knight to that square.

**public void** playMusic(String filename)

Plays the sound file with the specified filename.

**public void** changeStyle(String filename)

Changes the image of the knight and the visited square markers

**public void** getHighScores(String[] names, **int**[] times)

Reads the high scores from a file into an array.

**public void** writeHighScores(String[] names, **int**[] times)

Updates the high score file.

**public void** displayHighScores(String[] names, **int**[] times)

Displays the high scores in the table on the right of the screen (see screen layout)