Coms 262 The TourBoard Class Spring 2018

Introduction

A `TourBoard` object represents the rectangular board used to solve the Knight's Tour Problem.

Data Members

- `int m_rows` – the number of rows in the board.
- `int m_cols` – the number of columns in the board.
- `int m_num_of_moves` – the number of knight's moves made so far.
- `int m_board[15][15]` – the board, with a maximum of 15 rows and 15 columns. Each cell records the order in which that square was visited.

Public Member Functions

Constructors

- `TourBoard();`

  Constructs a `TourBoard` with 0 rows and 0 columns.

- `TourBoard(int r, int c);`

  Constructs a `TourBoard` with `r` rows and `c` columns, with each cell initialized to `0`.

Inspectors

- `int rows() const;`

  Returns the number of rows in the board.

- `int cols() const;`

  Returns the number of columns in the board.

- `bool occupied(Point p) const;`

  Returns `true` if the knight has already visited that square. It returns `false` otherwise.

- `bool solved() const;`

  Returns `true` if the puzzle has been solved. It returns `false` otherwise.

Mutators

- `void move(Point& p);`

  Increments the data member `m_num_of_moves`. Then updates the cell in the position indicated by the point `p` by assigning to it the number of moves.

- `void remove(Point& p);`

  Decrements the data member `m_num_of_moves`. Then updates the cell in the position indicated by the point `p` by assigning to it the value `0`.

Other Member Functions

- `void draw() const;`

  Displays the values in the cells in a rectangular array.

- `bool isLegal(Point& p) const;`

  Returns `true` if it is legal to move to the square indicated by the point `p`. Returns `false` otherwise.