```c
/*
 * File:   minimax.h
 * Author: Taylor Cowley
 *
 * Created on May 28, 2015, 12:40 PM
 */

#ifndef MINIMAX_H
#define MINIMAX_H

#include <stdint.h>
#include <stdbool.h>


// Defines the boundaries of the tic-tac-toe board.
#define MINIMAX_BOARD_ROWS 3
#define MINIMAX_BOARD_COLUMNS 3

// These are the values in the board to represent who is occupying what square.
#define MINIMAX_USED_SQUARE 3           // Used when creating new board states.
#define MINIMAX_PLAYER_SQUARE 2
#define MINIMAX_OPPONENT_SQUARE 1
#define MINIMAX_EMPTY_SQUARE 0

// Scoring for minimax.
#define MINIMAX_PLAYER_WINNING_SCORE 50
#define MINIMAX_OPPONENT_WINNING_SCORE -50
#define MINIMAX_DRAW_SCORE 0
#define MINIMAX_NOT_ENDGAME -1 // Not an end-game.
#define MINIMAX_INVALID_MOVE -127    //A semi-arbitrary value. No real score should be this.


// Boards contain just an array of squares. I used a struct to provide additional abstraction
// in case I wanted to add something to the board type.
typedef struct {
    int8_t squares[MINIMAX_BOARD_ROWS][MINIMAX_BOARD_COLUMNS];  // State of game as passed in.
} minimax_board_t;

// A move is simply a row-column pair.
typedef struct {
    uint8_t row;
    uint8_t column;
} minimax_move_t;

// Define a score type.
typedef int16_t minimax_score_t;

// This routine is not recursive but will invoke the recursive minimax function.
// It computes the row and column of the next move based upon:
// the current board,
// the player. true means the computer is X. false means the computer is O.
void minimax_computeNextMove(minimax_board_t* board, bool player, uint8_t* row, uint8_t* column);

// Determine that the game is over by looking at the score.
bool minimax_isGameOver(minimax_score_t score);

// Returns the score of the board, based upon the player.
```

```
int16_t minimax_computeBoardScore(minimax_board_t* board, bool player);

// Init the board to all empty squares.
void minimax_initBoard(minimax_board_t* board);

//This prints out the board. Woopdee doo. if sanitize is true, then it makes X and O
void minimax_print_board(minimax_board_t* board, bool sanitize);


#endif  /* MINIMAX_H */
```