```
/************************************************
 * tictac.c
 * Doug Lloyd
 * September 22, 2010
 *
 * A working, text-based, Tic-Tac-Toe game
 ************************************************/

/* Header files */
#include <stdio.h>
#include <cs50.h>

/* Global variables and constants */
#define DIM 3
#define MAX_SQ 9
#define BLANK '_'
char ttt_board[DIM][DIM];

/* Function Declarations */
void init_board();
bool is_valid_move(int sq);
void make_move(char c, int sq);
void print_board();
bool game_over();
bool check_row(int row);
bool check_col(int col);
bool check_diag(int diag);
bool all_full();

/* Function Definitions */
int main(int argc, char *argv[]) {
  init_board();
  print_board();
  int sq = 0;
  char c = BLANK;
  while(!game_over()) {
    do {
      printf("Enter your square (1-9): ");
      sq = GetInt();
    } while((sq < 1) || (sq > MAX_SQ) || !is_valid_move(sq));
    do {
      printf("Enter your letter (X/O): ");
      c = GetChar();
    } while((c != 'X') && (c != 'O'));
    make_move(c, sq);
    print_board();
  }
  printf("\nGame Over!\n");
  return 0;
}

void init_board() {
  for(int i = 0; i < DIM; i++)
    for(int j = 0; j < DIM; j++)
      ttt_board[i][j] = BLANK;
  return;
}

bool is_valid_move(int sq) {
  int usable_num = (sq - 1);
  int row = usable_num / DIM;
  int col = usable_num % DIM;

  if(ttt_board[row][col] == BLANK)
```

```c
    return true;
  else
    return false;
}

void make_move(char c, int sq) {
  int usable_num = (sq - 1);
  int row = usable_num / DIM;
  int col = usable_num % DIM;

  ttt_board[row][col] = c;
  return;
}

void print_board() {
  printf("\n\n\n");
  printf("-------------\n");
  for(int j = 0; j < DIM; j++) {
    printf("|");
    for(int k = 0; k < DIM; k++)
      printf(" %c |", ttt_board[j][k]);
    printf("\n-------------\n");
  }
  return;
}

bool game_over() {
  if(check_row(0) || check_row(1) || check_row(2))
    return true;
  if(check_col(0) || check_col(1) || check_col(2))
    return true;
  if(check_diag(1) || check_diag(2))
    return true;
  if(all_full())
    return true;
  return false;
}

bool check_row(int row) {
  if(ttt_board[row][0] != ttt_board[row][1])
    return false;
  if(ttt_board[row][1] != ttt_board[row][2])
    return false;
  if(ttt_board[row][0] == BLANK)
    return false;
  return true;
}

bool check_col(int col) {
  if(ttt_board[0][col] != ttt_board[1][col])
    return false;
  if(ttt_board[1][col] != ttt_board[2][col])
    return false;
  if(ttt_board[0][col] == BLANK)
    return false;
  return true;
}

bool check_diag(int diag) {
  if(diag == 1) {
    if(ttt_board[0][0] != ttt_board[1][1])
      return false;
    if(ttt_board[1][1] != ttt_board[2][2])
      return false;
  } else {
    if(ttt_board[0][2] != ttt_board[1][1])
      return false;
    if(ttt_board[1][1] != ttt_board[2][0])
      return false;
  }
  if(ttt_board[1][1] == BLANK)
    return false;
  return true;
}

bool all_full() {
  for(int i = 0; i < DIM; i++)
    for(int j = 0; j < DIM; j++)
      if(ttt_board[i][j] == BLANK)
        return false;
  return true;
}
```