SAIKOROAn engine for the game Saikoro

1. Here we describe a program for playing the Saikoro game. We'll write it as a vanilla C program that looks like this:

⟨ Header files to include 2 ⟩⟨ Global variables 5 ⟩⟨ Functions 10 ⟩⟨ The main program 9 ⟩

2. We're gonna need the standard header files, plus a couple more for generating random numbers.

```
⟨ Header files to include 2 ⟩ ≡
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
```

This code is used in section 1.

3. I don't really expect this program to work on anything else than the standard 8x8 board, but it's always nice to define these kinds of constants:

```
#define ROWS 8
#define COLS 8
```

4. We'll represent the state of the game by an 8x8 array, each entry of which holds either a number from 1 to 6, or 0 if a player has previously visited that square. We'll also keep track of each player's position.

5.

```
⟨ Global variables 5 ⟩ ≡
  struct player_position {
    int row;
    int col;
  };
  struct {
    int board[ROWS][COLS];
    struct player_position black, white;
  } game;
```

See also sections 7 and 8.

This code is used in section 1.

6. The positions of white and black will be marked on the board with the 'W' and 'B' characters.

```
#define WHITE 'W'
#define BLACK 'B'
```

7. The player whose turn it is.

```
⟨ Global variables 5 ⟩ +≡
  char PLAYING = BLACK;
```

8. I anticipate that we will eventually write different playing engines, each defined by a different strategy. That's why I begin by defining three different strategies:

```
⟨ Global variables 5 ⟩ +≡
  enum strategy_t {
    RANDOM, IMMEDIATE, SEARCH
  };
  enum strategy_t STRATEGY = IMMEDIATE;
```

9.

```
⟨ The main program 9 ⟩ ≡
  int main( )
  {
    init_game( );
    while (1) {
      show_board( );
      if (¬playing_can_move( )) break;
      if (BLACK ≡ PLAYING) {
        prompt_for_move( );
        PLAYING = WHITE;
      }
      else {
        find_best_move(STRATEGY);
        PLAYING = BLACK;
      }
    }
    announce_winner( );
  }
```

This code is used in section 1.

10. We initialize the game by rolling the dice and placing the players.

```
⟨ Functions 10 ⟩ ≡
  void init_game( )
  {
    int row, col;
    for (row = 0; row < ROWS; row++) {
      for (col = 0; col < COLS; col++) {
        game.board[row][col] = roll_die( );
      }
    }
    game.board[1][1] = 0;
    game.board[6][6] = 0;
    game.white.row = game.white.col = 1;
    game.black.row = game.black.col = 6;
  }
  int roll_die( )
  {
    return ((double) rand( )/(RAND_MAX + 1.)) * 6 + 1;
  }
```

See also sections 11 and 12.

This code is used in section 1.

11. Print the board to the console.

```
⟨ Functions 10 ⟩ +≡
  void show_board()
  {
    char board_as_string[] = "+-----+-----+-----+-----+-----+-----+--\
        ---+-----+\n""|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣\
        ␣␣␣|␣␣␣␣␣|\n""|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣\
        ␣␣␣|␣␣␣␣␣|\n""|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣\
        ␣␣␣|␣␣␣␣␣|\n""+-----+-----+-----+-----+-----+-----+--\
        ---+-----+\n""|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣\
        ␣␣␣|␣␣␣␣␣|\n""|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣\
        ␣␣␣|␣␣␣␣␣|\n""|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣\
        ␣␣␣|␣␣␣␣␣|\n""+-----+-----+-----+-----+-----+-----+--\
        ---+-----+\n""|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣\
        ␣␣␣|␣␣␣␣␣|\n""|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣\
        ␣␣␣|␣␣␣␣␣|\n""|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣\
        ␣␣␣|␣␣␣␣␣|\n""+-----+-----+-----+-----+-----+-----+--\
        ---+-----+\n""|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣\
        ␣␣␣|␣␣␣␣␣|\n""|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣\
        ␣␣␣|␣␣␣␣␣|\n""|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣\
        ␣␣␣|␣␣␣␣␣|\n""+-----+-----+-----+-----+-----+-----+--\
        ---+-----+\n""|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣\
        ␣␣␣|␣␣␣␣␣|\n""|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣\
        ␣␣␣|␣␣␣␣␣|\n""|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣\
        ␣␣␣|␣␣␣␣␣|\n""+-----+-----+-----+-----+-----+-----+--\
        ---+-----+\n""|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣\
        ␣␣␣|␣␣␣␣␣|\n""|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣\
        ␣␣␣|␣␣␣␣␣|\n""|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣\
        ␣␣␣|␣␣␣␣␣|\n""+-----+-----+-----+-----+-----+-----+--\
        ---+-----+\n""|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣\
        ␣␣␣|␣␣␣␣␣|\n""|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣\
        ␣␣␣|␣␣␣␣␣|\n""|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣\
        ␣␣␣|␣␣␣␣␣|\n""+-----+-----+-----+-----+-----+-----+--\
        ---+-----+\n""|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣\
        ␣␣␣|␣␣␣␣␣|\n""|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣\
        ␣␣␣|␣␣␣␣␣|\n""|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣␣␣␣|␣␣\
        ␣␣␣|␣␣␣␣␣|\n""+-----+-----+-----+-----+-----+-----+-----+-----+\n";
    int row, col;
    for (row = 0; row < ROWS; row++) {
      for (col = 0; col < COLS; col++) {
        if (game.board[row][col] > 0) {
          board_as_string[two_to_one_dim(row, col)] = '0' + game.board[row][col];
        }
      }
    }
    printf("%s", board_as_string);
  }
  int two_to_one_dim(int row, int col)
  {
    return 1503 − (row ∗ 200) + 6 ∗ col;
  }
```

12.

```
⟨ Functions 10 ⟩ +≡
  int playing_can_move( )
  {
    return 0;
  }
  void prompt_for_move( )
  {
    return;
  }
  void find_best_move(int strategy)
  {
    return;
  }
  void announce_winner( )
  {
    return;
  }
```

⟨ Functions 10, 11, 12 ⟩ Used in section 1.
⟨ Global variables 5, 7, 8 ⟩ Used in section 1.
⟨ Header files to include 2 ⟩ Used in section 1.
⟨ The main program 9 ⟩ Used in section 1.