

```

1  /*
2  PRP2-1 Aufgabe 1.2
3  Name: Malte Müller, Fabian Liebold
4  Date: 18.10.2018
5  */
6
7  #define _CRT_SECURE_NO_WARNINGS
8  #define MAX 100 //define max lenght of command string
9  #define SIZE 8 // length and width of playground
10
11 #include <stdio.h>
12 #include <string.h>
13
14 // Prototypes:
15 void init_playground(char playground[SIZE][SIZE]);
16 void print_playground(char playground[SIZE][SIZE]);
17 void play_str(char playground[SIZE][SIZE], char partie_copy[MAX]);
18 void update_playground(int counter, char playground[SIZE][SIZE], char figure,
19 char old_x, char old_y, char new_x, char new_y);
20 int mapping(char input);
21 void play_input(char playground[SIZE][SIZE]);
22
23 int main(void) {
24
25     char partie[MAX] =
26         "e2-e4/e7-e5/Sg1-f3/Sb8-c6/Lf1-c4/Lf8-c5/Sf3xe5/Sc6xe5/De1-e3/Lc5-e3/"; //
27     given partie string
28     char partie_copy[MAX];
29     strcpy(partie_copy, partie); // Create copy for strtok
30
31     //Print to console:
32     printf("*** Schach - Nachspielen einer Partie ***\n\n");
33     printf("Folgende Parite wird gespielt:\n");
34     printf("%s\n", partie);
35     printf("Die Zuege sind beginnend mit weiss abwechselt notiert.\nDie weissen
36     Figuren im Spielfeld mit Grossbuchstaben bezeichnet.\n");
37
38     //Initialize playground array and print it to console
39     char playground[SIZE][SIZE];
40     init_playground(playground);
41     print_playground(playground);
42
43     play_str(playground, partie_copy); // play the commands from the string
44
45     char flag = 'N';
46     printf("Wollen Sie die Partie weiterspielen? [J/N]:");
47     scanf("%c", &flag);
48     if(flag == 'j' || flag == 'J') play_input(playground); // check if user wants
49     to play further
50
51     printf("\nSpiel beendet.\n");
52
53     system("PAUSE");
54     return 1;
55 }
56
57 /*****
58 *****
59 Function initializes playground array.
60 Parameters:
61     char playground[SIZE][SIZE]
62 Returns:
63     void.
64 */
65 void init_playground(char playground[SIZE][SIZE]) {
66
67     playground[0][0] = 'T';
68     playground[0][1] = 'S';

```

```

69     playground[0][2] = 'L';
70     playground[0][3] = 'K';
71     playground[0][4] = 'D';
72     playground[0][5] = 'L';
73     playground[0][6] = 'S';
74     playground[0][7] = 'T';
75
76     playground[7][0] = 't';
77     playground[7][1] = 's';
78     playground[7][2] = 'l';
79     playground[7][3] = 'k';
80     playground[7][4] = 'd';
81     playground[7][5] = 'l';
82     playground[7][6] = 's';
83     playground[7][7] = 't';
84
85     for (int i = 0; i < SIZE; i++) {
86         playground[1][i] = 'B';
87     }
88
89     for (int i = 0; i < SIZE; i++) {
90         playground[6][i] = 'b';
91     }
92
93
94     for (int i = 2; i < 6; i++) {
95         for (int j = 0; j < SIZE; j++) {
96             if ((i % 2 == 0 && j % 2 == 0) || i % 2 == 1 && j % 2 == 1) {
97                 playground[i][j] = ' ';
98             }
99             else playground[i][j] = 177;
100         }
101     }
102 }
103
104 /*****
105 *****
106 Function prints the playground to the console.
107 Parameters:
108     char playground[SIZE][SIZE]
109 Returns:
110     void.
111 */
112 void print_playground(char playground[SIZE][SIZE]) {
113
114     printf("\nSpielbrett\n    |\n");
115     for (int i = 7; i >= 0; i--) {
116         printf("    %i | %c %c %c %c %c %c %c %c \n", i+1, playground[i][0],
117             playground[i][1], playground[i][2],
118             playground[i][3], playground[i][4], playground[i][5], playground[i][6],
119             playground[i][7]);
120     }
121     printf("----+-----\n    | a b c d e f g h\n\n");
122 }
123
124 /*****
125 *****
126 Function, which handles the commands of the given string.
127 Parameters:
128     char playground[SIZE][SIZE]
129     char partie_copy[MAX] - given command string.
130 Returns:
131     void.
132 */
133 void play_str(char playground[SIZE][SIZE], char partie_copy[MAX]) {
134
135     char delimiter[] = "/"; // set delimiter
136     char* token_ptr = strtok(partie_copy, delimiter); // Create pointer, which
137     points on the beginning of each word
138     char fig = ' ';
139
140     int counter = 1;
141     int play_flag = 1;

```

```

137
138 while (token_ptr != 0) { // While a command is available
139
140     system("PAUSE"); // Wait until the user presses 'return' to continue
141
142     printf("\n%i.Zug: %s\n", counter, token_ptr); // Print current command
143
144     if (strlen(token_ptr) == 5) { // Check the length of the command
145         if (token_ptr[2] == '-') { // differentiate if it's "Setzzug" or
            "Schlagzug"
146             printf("Dieser Zug ist ein Setzzug:\n");
147         }
148         else {
149             printf("Dieser Zug ist ein Schlagzug:\n");
150         }
151
152         if (counter % 2 == 1) fig = 'B';
153         else fig = 'b';
154
155         update_playground(counter, playground, fig, token_ptr[0], token_ptr[1],
            token_ptr[3], token_ptr[4]); // update playground
156     }
157     else if (strlen(token_ptr) == 6) {
158         if (token_ptr[3] == '-') {
159             printf("Dieser Zug ist ein Setzzug:\n");
160         }
161         else {
162             printf("Dieser Zug ist ein Schlagzug:\n");
163         }
164         if (counter % 2 == 1) fig = token_ptr[0];
165         else fig = token_ptr[0] + 32; // If counter is odd -> get small
            character by adding 32 to the ascii value
166
167         update_playground(counter, playground, fig, token_ptr[1], token_ptr[2],
            token_ptr[4], token_ptr[5]); // update playground
168     }
169
170     print_playground(playground); // Print updated playground-array to console
171
172     token_ptr = strtok(NULL, delimiter); // reset token pointer
173     counter++; // Increase counter
174 }
175 }
176
177 /*****
178 *****
179 Function updates the playground-array.
180 Parameters:
181     int counter - counts turns.
182     char playground[SIZE][SIZE]
183     char figure - figure, which should be moved.
184     char old_x - start position.
185     char old_y
186     char new_x - destination postion.
187     char new_y
188 Returns:
189     void.
190 */
191 void update_playground(int counter, char playground[SIZE][SIZE], char figure,
    char old_x, char old_y, char new_x, char new_y) {
192
193     playground[(int)new_y - 49][mapping(new_x)] = figure; // Get the current
        figure of the start-position
194
195                                     //(-49 because Number is a character and
        has to be converted to int)
196
197     if ((mapping(old_x) % 2 == 0 && ((int)old_y - 49) % 2 == 0) ||
        (mapping(old_x) % 2 == 1 && ((int)old_y - 49) % 2 == 1)) { // check the
        pattern
198         playground[(int)old_y - 49][mapping(old_x)] = ' ';
199     }
200     else playground[(int)old_y - 49][mapping(old_x)] = 177;
201

```

```

202 }
203
204 /*****
*****
205 Function, which handles the commands of the given string.
206 Parameters:
207     char input - character, which should be mapped to corresponding int.
208 Returns:
209     int - corresponding int.
210 */
211 int mapping(char input) {
212     /*
213     Function maps the characters a to h to its corresponding numbers
214     */
215
216     int v = 0;
217
218     switch (input)
219     {
220     case 'a':
221         v = 0;
222         break;
223     case 'b':
224         v = 1;
225         break;
226     case 'c':
227         v = 2;
228         break;
229     case 'd':
230         v = 3;
231         break;
232     case 'e':
233         v = 4;
234         break;
235     case 'f':
236         v = 5;
237         break;
238     case 'g':
239         v = 6;
240         break;
241     case 'h':
242         v = 7;
243         break;
244     default:
245         break;
246     }
247
248     return v;
249 }
250
251 /*****
*****
252
253 Fuction handles user input.
254 Parameters:
255     char playground[SIZE][SIZE]
256 Returns:
257     void.
258 */
259 void play_input(char playground[SIZE][SIZE]) {
260     /*
261     Fuction handles user input.
262     */
263     char old_value[3]; // Create String of length of 3 "XX\0"
264     char new_value[3]; // Create String of length of 3 "XX\0"
265     int counter = 1;
266     char fig;
267     int x, y;
268
269     while (1) { // Create a loop
270
271         printf("Beenden der Partie durch Eingabe eines 00!\n");
272         printf("Eingabe Ausgangsfeld [SpalteZeile]: ");

```

```
273     scanf("%s", old_value);
274
275     if (old_value[0] == '0' && old_value[1] == '0') return; // Return to main,
    if user inputs "00"
276
277     printf("Eingabe Zielfeld [SpalteZeile]: ");
278     scanf("%s", new_value);
279
280     x = (int)old_value[1] - 49; // Get x value by converting the character to
    int and subtract 49 from ascii value
281     y = mapping(old_value[0]); // Get y value by mapping the character to its
    number
282
283     fig = playground[x][y]; //get selected figure
284
285     update_playground(counter, playground, fig, old_value[0],
286         old_value[1], new_value[0], new_value[1]); // update playground
287
288     print_playground(playground); // print playground
289
290     counter++; // Increase counter
291 }
292
293 }
```