

CSE3013 (컴퓨터공학 설계 및 실험 I)

WEB-3 결과 보고서

서강대학교 컴퓨터공학과 박수현 (20181634)

서강대학교 컴퓨터공학과

1 소스 코드

이 코드는 <https://shiftpsh.com/web3/> 에서 실행시켜 볼 수 있다.

```
1  <!DOCTYPE html>
2  <html>
3
4  <head>
5  <title>15 Puzzle Game</title>
6  <meta charset="utf-8">
7  <script language="JavaScript">
8      var completed = true; // 퍼즐이 완성되었는지에 대한 여부
9      // 문자열(str)을 구분자(sep)로 끊은 토큰들의 리스트를 생성
10     // sep: 구분자(seprator)
11     function tokenize(sep, str) {
12         tokens = new Array(); // 토큰 리스트
13         i = 0;
14         while (1) {
15             idx = str.indexOf(sep);
16             if (idx == -1) {
17                 if (str.length > 0) {
18                     tokens[i] = str;
19                 }
20                 break;
21             }
22             tokens[i++] = str.substring(0, idx); // 새로운 토큰 추가
23             str = str.substr(idx + 1); // 다음 토큰을 찾을 문자열
24         }
25         return tokens;
26     }
```

```

27
28 // 퍼즐판에서의 x 좌표
29 function getX(idy) {
30     var rest = idy - Math.floor(idy / 4) * 4;
31     return (rest == 0) ? 4 : rest;
32 }
33
34 // 퍼즐판에서의 y 좌표
35 function getY(idy) {
36     return Math.floor((idy - 1) / 4) + 1;
37 }
38
39 // 퍼즐판에서의 좌표(x,y)가 몇 번째 칸인지를 계산
40 function getIndex(x, y) {
41     return x + (y - 1) * 4;
42 }
43
44 // Get value at given index
45 function getValueAt(idy) {
46     var currentSrc = tokenize("/", document.images[idy - 1].src);
47     return parseInt(tokenize(".", currentSrc[currentSrc.length - 1])[0]);
48 }
49
50 // x(혹은 y) 좌표를 입력받아 이동할 수 있는 새로운 좌표를 구함
51 function newDirection(pos) {
52     var dir;
53     if ((pos == 2) || (pos == 3)) dir = (Math.floor(Math.random() + 0.5) == 0)
54         ? -1 : 1;
55     else dir = (pos == 1) ? 1 : -1;
56     return (pos + dir);
57 }
58
59 // 퍼즐 조각이 새로 이동할 수 있는 칸을 계산
60 function newIndex(idy) {
61     var x, y;
62     x = getX(idy);
63     y = getY(idy);
64     if (Math.floor(Math.random() + 0.5) == 0) x = newDirection(x);
65     else y = newDirection(y);
66     return getIndex(x, y);
67 }

```

```
67
68 // 퍼즐이 완성되었는지에 대한 여부
69 function isComplete() {
70     // 배열 document.images[]의 원소들(각 원소에 할당된 퍼즐 조각 그림들)이
71     //   ↳ 오름차순으로 차례대로 정렬되어 있고
72     // completed가 false이면,
73     // 퍼즐이 완성된 것이다.
74     if (completed) return false;
75     // Check if pieces are sorted
76     for (var i = 2; i <= 16; i++) {
77         if (getValueAt(i) != getValueAt(i - 1) + 1) return false;
78     }
79     return true;
80 }
81
82 // 퍼즐을 섞음
83 function shuffle() {
84     var puzzles = new Array(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
85     //   ↳ 16);
86     var empty = 15; // Position of empty cell
87     // iter: 퍼즐 조각을 섞는 횟수
88     iter = Math.floor(Math.random() * 200 + 0.5) + 100;
89     // puzzles 배열의 원소들을 섞는다.
90     for (var i = 1; i < iter; i++) {
91         // puzzles[] 배열의 임의로 선택된 퍼즐판상에서의 인접한 두 퍼즐조각을 서로
92         //   ↳ 바꾼다.
93         // Swap newIndex value with empty value
94         var fromIndex = newIndex(empty + 1) - 1;
95         puzzles[empty] = puzzles[fromIndex];
96         puzzles[fromIndex] = 16; // Since the value of the empty cell will be
97         //   ↳ always 16
98         empty = fromIndex;
99     }
100     // 섞인 퍼즐 조각들을 배열 document.images[]에 각각 할당
101     for (var i = 0; i < 16; i++)
102         document.images[i].src = puzzles[i].toString() + ".gif";
103     // 퍼즐을 섞으면 게임이 새로 시작되므로 completed 변수를 false로 초기화
104     completed = false;
105 }
106
107 // 퍼즐 조각을 이동
108 function movePiece(idx) {
```

```

104 // 퍼즐판에서 idx 번째 칸에 있는 퍼즐조각에 대한 퍼즐 조각 번호를 얻어낸다.
105 // 만약 그 번호가 16이면 blank를 의미하므로 더 이상의 작업이 필요없다.
106 if (getValueAt(idx) == 16) return;
107 // x: 선택된 퍼즐 조각의 퍼즐판상에서의 x 좌표
108 var x = getX(idx);
109 // y: 선택된 퍼즐 조각의 퍼즐판상에서의 y 좌표
110 var y = getY(idx);
111 // flag: 이동이 가능하면 1, 아니면 0
112 var flag = false;
113 // midx: 이동가능한 칸의 번호
114 var midx;
115 // 선택된 퍼즐 조각의 상하좌우 방향으로 바로 인접한 칸에 대하여
116 var dx = [1, -1, 0, 0];
117 var dy = [0, 0, 1, -1];
118 // 비어있는 칸을 찾는다(해당 칸의 퍼즐조각 번호가 16이면 비어있는 칸).
119 for (var d = 0; d < 4; d++) {
120     // Check coordinates (nx, ny)
121     var nx = x + dx[d];
122     var ny = y + dy[d];
123     // Boundary check, 1 <= nx <= 4; 1 <= ny <= 4.
124     if (0 >= nx || nx > 4) continue;
125     if (0 >= ny || ny > 4) continue;
126     // Empty cell check
127     if (getValueAt(getIndex(nx, ny)) == 16) {
128         flag = true;
129         midx = getIndex(nx, ny);
130         break;
131     }
132 }
133 // 비어있는 칸이 존재하면
134 if (flag) {
135     // 선택된 칸의 퍼즐 조각을 그 칸으로 이동시킨다.
136     var temp = document.images[idx - 1].src;
137     document.images[idx - 1].src = document.images[midx - 1].src;
138     document.images[midx - 1].src = temp;
139 }
140 // 퍼즐이 완성되었는지 확인하여
141 if (isComplete()) {
142     // 퍼즐이 완성된 상태면,
143     // "Congratulation!" 메시지를 경고창(alert box)으로 띄어주고 completed
    ↪ 변수를 true로 세팅한다.

```

```
144         alert("Congratulation!");
145         completed = true;
146     }
147 }
148 </script>
149
150 </head>
151
152 <body bgcolor="silver" text="black" link="#0000EE" vlink="#551A8B" alink="red">
153 <h2 align="center">15 Puzzle</h2>
154 <div align="center">
155     <table border>
156         <tr>
157             <td width="50%" align="center">
158                 <script language="JavaScript">
159                     with (window.document) {
160                         open();
161                         writeln('<table border=1 cellpadding=0 cellspacing=1>');
162                         for (var i = 1; i < 17; i++) {
163                             if (i == 1 || i == 5 || i == 9 || i == 13) {
164                                 writeln('<tr>');
165                             }
166                             writeln('    <td width=49 height=49>');
167                             writeln('        <a href=JavaScript:movePiece(' + i +
168                                 ' → ');
169                             writeln('        <img src=', i, '.gif border=0 width=49
170                                 ' → height=49 name=i', i, '></a>');
171                             writeln('    </td>');
172                             if (i == 4 || i == 8 || i == 12 || i == 16) {
173                                 writeln('</tr>');
174                             }
175                         }
176                         writeln('</table>');
177                         close();
178                     }
179                 </script>
180             </td>
181         </tr>
182     </table>
183 </div>
184 <p align="center"><br></p>
```

```
183 <form method="get">
184   <p align="center">
185     <input type="button" value="시작" onclick="shuffle()"></p>
186 </form>
187 </body>
188
189 </html>
```