

```
1  var tiles = [];
2  var freeCell = {
3    y: 3,
4    x: 3
5  };
6  /* для избежания проверки на собранность
7   во время перемешивания */
8  var shuffled = false;
9
10 function createCellNull() {
11   var cell = document.createElement("div");
12   cell.classList.add("field__cell", "field__cell--null");
13   return cell;
14 }
15
16 function setCellOffset(cell) {
17   /* + "px" необходимо, так как это значение свойства CSS */
18   cell.style.left = (15 + (15 + 81.25) * cell.x) + "px";
19   cell.style.top = (15 + (15 + 81.25) * cell.y) + "px";
20 }
21
22 function appendCell(cell) {
23   var field = document.getElementById("field");
24   field.appendChild(cell);
25 }
26
27 function createField() {
28   var x, y, cell;
29   for (y = 0; y < 4; ++y) {
30     for (x = 0; x < 4; ++x) {
31       cell = createCellNull();
32       cell.y = y;
33       cell.x = x;
34       setCellOffset(cell);
35       appendCell(cell);
36     }
37   }
38 }
39
40 function createCellTile() {
41   var cell = document.createElement("div");
42   cell.classList.add("field__cell", "field__cell--tile");
43   return cell;
44 }
45
```

```
46 function createTiles() {
47     var x, y, cell, n;
48     for (y = 0; y < 4; ++y) {
49         for (x = 0; x < 4; ++x) {
50             n = y * 4 + x + 1;
51             /* всего костяшек должно быть 15 */
52             if (n < 16) {
53                 cell = createCellTile();
54                 cell.y = y;
55                 cell.x = x;
56                 cell.innerHTML = n;
57                 setCellOffset(cell);
58                 appendCell(cell);
59                 /* добавляет костяшку в массив */
60                 tiles.push(cell);
61             }
62         }
63     }
64 }
65
66 function between(a, b, t) {
67     return a <= t && t <= b || b <= t && t <= a;
68 }
69
70 function checkVictory() {
71     var i, n;
72     for (i = 0; i < tiles.length; ++i) {
73         n = tiles[i].y * 4 + tiles[i].x + 1;
74         /* нестрогое сравнение, так как innerHTML -- строка */
75         if (tiles[i].innerHTML != n) return;
76     }
77     document.getElementById("modal").classList.add("modal--visible");
78 }
79
```

```
80 function tileClick(event) {
81     var bar = event.target,
82         i, tile;
83     /* запоминаем старые координаты каждой ячейки */
84     var oldX = bar.x,
85         oldY = bar.y;
86     if (bar.y == freeCell.y) {
87         for (i = 0; i < tiles.length; ++i) {
88             tile = tiles[i];
89             if (tile.y == bar.y && between(bar.x, freeCell.x, tile.x)) {
90                 if (bar.x < freeCell.x) tile.x += 1;
91                 else tile.x -= 1;
92                 setCellOffset(tile);
93             }
94         }
95         freeCell = {
96             y: oldY,
97             x: oldX
98         };
99     } else if (bar.x == freeCell.x) {
100         for (i = 0; i < tiles.length; ++i) {
101             tile = tiles[i];
102             if (tile.x == bar.x && between(bar.y, freeCell.y, tile.y)) {
103                 if (bar.y < freeCell.y) tile.y += 1;
104                 else tile.y -= 1;
105                 setCellOffset(tile);
106             }
107         }
108         freeCell = {
109             y: oldY,
110             x: oldX
111         };
112     }
113     /* если перемешание закончилось,
114     проверяем, собрана ли головоломка */
115     if (shuffled) {
116         checkVictory();
117     }
118 }
119
120 function animateTiles() {
121     var i;
122     for (i = 0; i < tiles.length; ++i) {
123         tiles[i].addEventListener("click", tileClick);
124     }
125 }
126
```

```
127 function shuffleTiles() {  
128     var i, index;  
129     for (i = 0; i < 1000; ++i) {  
130         index = Math.floor(Math.random() * tiles.length);  
131         tiles[index].click();  
132     }  
133     shuffled = true;  
134 }  
135  
136 createField();  
137 createTiles();  
138 animateTiles();  
139 shuffleTiles();
```