

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-model href="http://docbook.org/xml/5.1/rng/docbook.rng"
schematypens="http://relaxng.org/ns/structure/1.0"?> <?xml-model
href="http://docbook.org/xml/5.1/sch/docbook.sch" type="application/xml"
schematypens="http://purl.oclc.org/dsdl/schematron"?> <article
xmlns="http://docbook.org/ns/docbook"
xmlns:xlink="http://www.w3.org/1999/xlink" version="5.1">
<info>
<title>Документация к проекту "Сапёр"</title>
<author>
<personname>
<firstname>Кирилл</firstname>
<surname>Колодий</surname>
</personname>
</author>
<copyright>
<year>2021</year>
<holder>Кирилл Колодий</holder>
</copyright>
</info>
<chapter>
<title>Описание проекта</title>
<para>«Сапёр» (англ. Minesweeper) — компьютерная игра-головоломка.</para>
<section>
<title>Правила игры</title>
<para>1. Игрок открывает ячейки, стараясь не
открыть ячейку с миной. Открыв ячейку с миной, он проигрывает.</para>
<para>2. Если под открытой ячейкой мины нет, то в ней появляется число,
показывающее, сколько ячеек, соседствующих с только что открытой,
заминировано используя эти числа, игрок пытается рассчитать расположение
мин, однако иногда даже в середине и в конце игры некоторые ячейки всё же
приходится открывать наугад.</para>
<para>3. Если под соседними ячейками
тоже нет мин, то открывается некоторая не заминированная область до ячеек,
в которых есть цифры. </para>
<para>4. Заминированные ячейки игрок может
пометить, чтобы случайно не открыть их.</para>
<para>5. Открыв все не заминированные ячейки, игрок выигрывает.</para>
</section>
</chapter>
<chapter>
<title>Детали реализации</title>
<para>Особенности реализации отдельного функционала представлены ниже:</para>
<section>
<title>Функция "init"</title>
<para>Создадим плитки, мины и разместим их на экране.</para>
<example>
<programlisting><![CDATA[
function init() {
mines = 5;
remaining = mines;
revealed = 0;
status.innerHTML = 'Click on the tiles to reveal them';
for (let row = 0; row < rows; row++)
  for (let column = 0; column < columns; column++) {
    let index = row * columns + column;
    tile[row][column] = document.createElement('img');
    tile[row][column].src = 'img/hidden.png';
    tile[row][column].style = 'position:absolute;height:30px; width: 30px';
    tile[row][column].style.top = 150 + row * 30;
```

```

    tile[row][column].style.left = 50 + column * 30; tile[row]
[column].addEventListener('mousedown', click);
    tile[row][column].id = index;
    document.body.appendChild(tile[row][column]);
    picture[row][column] = 'hidden';
    board[row][column] = '';
}

let placed = 0;
while (placed < mines) {
    let column = Math.floor(Math.random() * columns);
    let row = Math.floor(Math.random() * rows);

    if (board[row][column] !== 'mine') {
        board[row][column] = 'mine';
        placed++;
    }
}

for (let column = 0; column < columns; column++)
    for (let row = 0; row < rows; row++) {
        if (check(row, column) !== 'mine') {
            board[row][column] = ((check(row + 1, column) === 'mine') | 0) +
((check(row + 1, column - 1) === 'mine') | 0) +
((check(row + 1, column + 1) === 'mine') | 0) +
((check(row - 1, column) === 'mine') | 0) +
((check(row - 1, column - 1) === 'mine') | 0) +
((check(row - 1, column + 1) === 'mine') | 0) +
((check(row, column - 1) === 'mine') | 0) +
((check(row, column + 1) === 'mine') | 0);
        }
    }
}
]]></programlisting>
</example>
</section>
<section>
<title>Функция "click"</title>
<para>Обработчик нажатия на плитку левой/правой кнопкой мыши.
В зависимости от нажатия кнопки изменяется состояние плитки.</para>
<example>
<programlisting><![CDATA[ function click(event) {
let source = event.target;
let id = source.id;
let row = Math.floor(id / columns);
let column = id % columns;

if (event.which === 3) {
    switch (picture[row][column]) {
        case 'hidden':
            tile[row][column].src = 'img/flag.png';
            remaining--;
            picture[row][column] = 'flag';
            break;
        case 'flag':
            tile[row][column].src = 'img/question.png';
            remaining++;

```

```

picture[row][column] = 'question';
break;
case 'question':
tile[row][column].src = 'img/hidden.png';
picture[row][column] = 'hidden';
break;
} event.preventDefault();
}
status.innerHTML = 'Mines remaining: ' + remaining;

if (event.which == 1 && picture[row][column] != 'flag') {
  if (board[row][column] == 'mine') {
    for (let row = 0; row < rows; row++)
    for (let column = 0; column < columns; column++) {
      if (board[row][column] == 'mine') {
        tile[row][column].src = 'img/mine.png';
      }
      if (board[row][column] != 'mine' && picture[row][column] == 'flag') {
        tile[row][column].src = 'img/misplaced.png';
      }
    }
    status.innerHTML = 'GAME OVER<br><br>Click here to restart';
  } else
  if (picture[row][column] == 'hidden') reveal(row, column);
}
if (revealed == rows * columns - mines)
status.innerHTML = 'YOU WIN!<br><br>Click here to restart';
}
]]></programlisting>
</example>
</section>
<section>
<title>Функция "reveal"</title>
<para>Будет вызывать себя рекурсивно, пока не будут обнаружены все
соседние нулевые плитки.</para> <example>
<programlisting><![CDATA[
function reveal(row, column) {
tile[row][column].src = 'img/' + board[row][column] + '.png';
if (board[row][column] != 'mine' && picture[row][column] == 'hidden')
  revealed++;
picture[row][column] = board[row][column];

if (board[row][column] == 0) {
  if (column > 0 && picture[row][column - 1] == 'hidden') reveal(row,
column - 1); if (column < (columns - 1) && picture[row][+column + 1] ==
'hidden') reveal(row, +column + 1); if (row < (rows - 1) && picture[+row
+ 1][column] == 'hidden') reveal(+row + 1, column);
  if (row > 0 && picture[row - 1][column] == 'hidden') reveal(row - 1,
column);
  if (column > 0 && row > 0 && picture[row - 1][column - 1] == 'hidden')
reveal(row - 1, column - 1); if (column > 0 && row < (rows - 1) &&
picture[+row + 1][column - 1] == 'hidden') reveal(+row + 1, column - 1);
if (column < (columns - 1) && row < (rows - 1) && picture[+row + 1]
[+column + 1] == 'hidden') reveal(+row + 1, +column + 1); if (column <
(columns - 1) && row > 0 && picture[row - 1][+column + 1] == 'hidden')
reveal(row - 1, +column + 1); }
}
]]></programlisting>
</example>

```

```
]]></programlisting>  
</example>  
</section>  
</chapter>  
</article>
```