

## Создание игры «Breakout» на JavaScript

*Семченко Регина Викторовна*

*Приамурский государственный университет имени Шолом-Алейхема*

*студент*

*Еровлев Павел Андреевич*

*Приамурский государственный университет имени Шолом-Алейхема*

*студент*

### Аннотация

В данной статье рассмотрен процесс написания собственной игры на языке JavaScript. Был изучен пошаговым методом код игры. Практическим результатом является собственная рабочая игра.

**Ключевые слова:** JavaScript, Breakout, игра, web-игра

## Creating game «Breakout» in JavaScript

*Semchenko Regina Viktorovna*

*Sholom-Aleichem Priamursky State University*

*student*

*Erovlev Pavel Andreevich*

*Sholom-Aleichem Priamursky State University*

*student*

### Abstract

This article describes the process of writing your own game in JavaScript. The game code was studied step by step. The bottom line is your own working game.

**Keywords:** JavaScript, Breakout, game, web game

Игра «Breakout» известна не только молодому поколению людей, ведь эта игра была выпущена в далеком 1976 году командой разработчиков компании “Atari”. Изначально это был игровой автомат с одной игрой «Breakout», в последующие года игра перешла на «Тетрис», компьютер, планшет, телефон. Суть данной игры заключается в том, чтобы сломать все ряды блоков одним шариком. В игре есть жизни, они теряются, когда мяч падает на «пол», поэтому у игрока есть доска, которой он управляет, а шарик от нее отскакивает и летит в обратную сторону. Игра подразумевает одиночный геймплей (рис.1) и игру на 2-ух игроков (рис.2). Процесс мультиплеерной игры заключается в победе одного из игроков, игра продолжается до тех пор, пока у одного из двух не закончатся жизни. Так же

в игре имеется возможность заработать очки, сбивая те самые блоки, один блок равен одному очку.

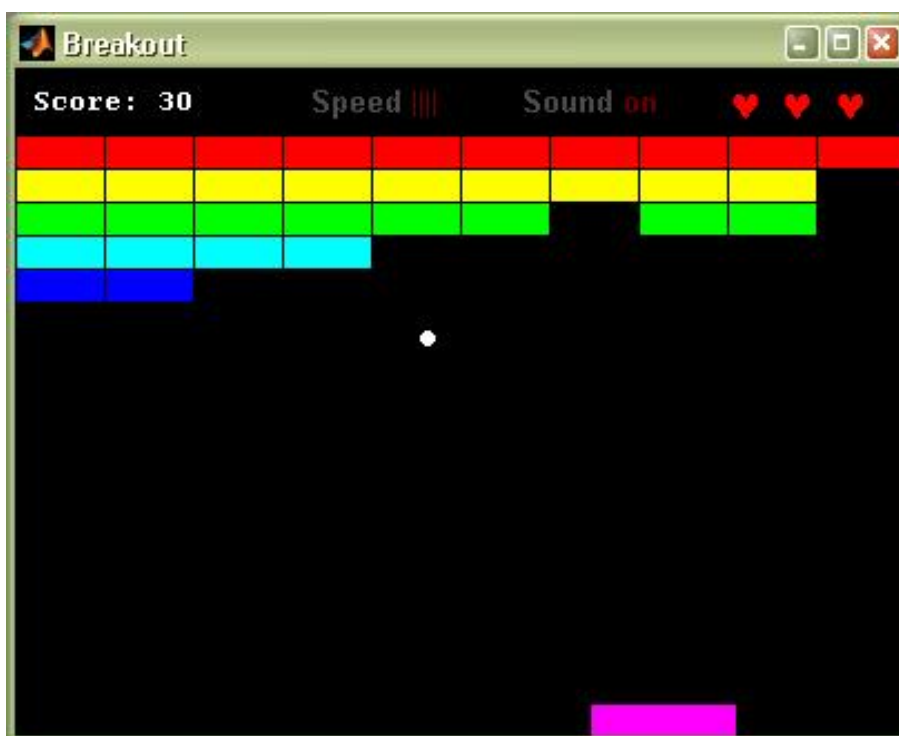


Рисунок 1 – Одиночная игра

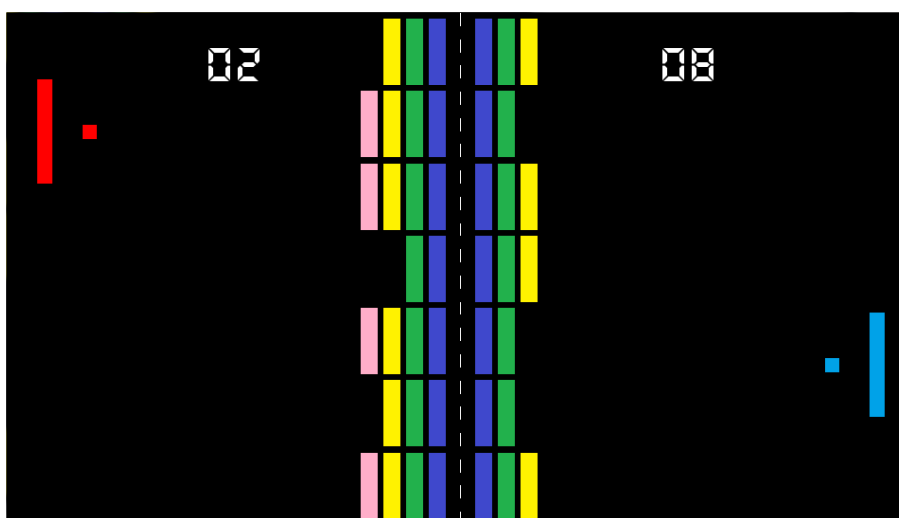


Рисунок 2 – мультиплеер

Цель данной статьи разработать собственную браузерную игру «Breakout» с одиночным режимом игры на языке программирования JavaScript.

В работе Беляева С.А. рассматриваются ключевые вопросы разработки однопользовательских двумерных игр на языке JavaScript[3]. А.В.Диков исследовал объект Math языка JavaScript, который обладает множеством математических свойств и методов. Таким образом открывается возможность использования языка веб программирования для решения математических задач[12]. Зудилова Т.В., Буркова М.Л в своем пособии рассматривают

необходимые для создания игры объекты `math` и `document`, а так же рассказывают о способе создания и вызова функций[5]. В своей работе Прохоренок Н., Дронов В. рассмотрели вопросы создания интерактивных Web-сайтов с помощью HTML, JavaScript[8]. При создании игры были изучены и прочитаны следующие источники. В учебном пособии Захаркина В. В. рассмотрены основы применения алгоритмического языка JavaScript при создании интерактивных веб-документов[4].

Для написания кода создадим файлы “breakout.html” и “js/game.js”. В html-файле пропишем стандартные мета тэги и тэг “canvas” с шириной 480 и высотой 320 пикселей(рис.3).

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>Breakout</title>
</head>
<body>

<canvas id="canvas" width="480" height="320"></canvas>

<script src="js/game.js"></script>
</body>
</html>
```

Рисунок 3 – Код в html-файле

Следующим шагом будет написание java кода, здесь не будет готовых картинок, а нарисуем свои блоки, доску, мяч. Создадим переменные игрового поля, счета, жизней, блоков и их размеров, радиус мяча, а так же укажем, что это 2d игра (рис.4)

```
var canvas = document.getElementById("canvas");
var ctx = canvas.getContext("2d");
var ballRadius = 10;
var x = canvas.width/2;
var y = canvas.height-30;
var dx = 2;
var dy = -2;
var paddleHeight = 10;
var paddleWidth = 75;
var paddleX = (canvas.width-paddleWidth)/2;
var rightPressed = false;
var leftPressed = false;
var brickRowCount = 5;
var brickColumnCount = 3;
var brickWidth = 75;
var brickHeight = 20;
var brickPadding = 10;
var brickOffsetTop = 30;
var brickOffsetLeft = 30;
```

```

var score = 0;
var lives = 3;

var bricks = [];
for(var c=0; c<brickColumnCount; c++) {
  bricks[c] = [];
  for(var r=0; r<brickRowCount; r++) {
    bricks[c][r] = { x: 0, y: 0, status: 1 };
  }
}

```

Рисунок 4 – Добавление переменных

Дальше сделаем возможность управления. Управляющие клавиши будут назначены на стрелки вправо и влево соответственно. Так же добавим возможность управлять мышкой (рис.5).

```

document.addEventListener("keydown", keyDownHandler, false);
document.addEventListener("keyup", keyUpHandler, false);
document.addEventListener("mousemove", mouseMoveHandler, false);

function keyDownHandler(e) {
  if(e.key == "Right" || e.key == "ArrowRight") {
    rightPressed = true;
  }
  else if(e.key == "Left" || e.key == "ArrowLeft") {
    leftPressed = true;
  }
}

function keyUpHandler(e) {
  if(e.key == "Right" || e.key == "ArrowRight") {
    rightPressed = false;
  }
  else if(e.key == "Left" || e.key == "ArrowLeft") {
    leftPressed = false;
  }
}

function mouseMoveHandler(e) {
  var relativeX = e.clientX - canvas.offsetLeft;
  if(relativeX > 0 && relativeX < canvas.width) {
    paddleX = relativeX - paddleWidth/2;
  }
}

```

Рисунок 5 – Назначение управления

Осталось написать функцию, которая позволит мячу не просто проходить сквозь блоки, а ломать их. В функции прописан код, где при столкновении с мячом блок разрушается и засчитывается одно очко, а при выбивании всех блоков выходит окно с сообщением о победе (рис.6).

```

function collisionDetection() {
  for(var c=0; c<brickColumnCount; c++) {

```



```
document.addEventListener("mousemove", mouseMoveHandler, false);

function keyDownHandler(e) {
    if(e.key == "Right" || e.key == "ArrowRight") {
        rightPressed = true;
    }
    else if(e.key == "Left" || e.key == "ArrowLeft") {
        leftPressed = true;
    }
}

function keyUpHandler(e) {
    if(e.key == "Right" || e.key == "ArrowRight") {
        rightPressed = false;
    }
    else if(e.key == "Left" || e.key == "ArrowLeft") {
        leftPressed = false;
    }
}

function mouseMoveHandler(e) {
    var relativeX = e.clientX - canvas.offsetLeft;
    if(relativeX > 0 && relativeX < canvas.width) {
        paddleX = relativeX - paddleWidth/2;
    }
}

function collisionDetection() {
    for(var c=0; c<brickColumnCount; c++) {
        for(var r=0; r<brickRowCount; r++) {
            var b = bricks[c][r];
            if(b.status == 1) {
                if(x > b.x && x < b.x+brickWidth && y > b.y && y < b.y+brickHeight) {
                    dy = -dy;
                    b.status = 0;
                    score++;
                    if(score == brickRowCount*brickColumnCount) {
                        alert("Вы победили, поздравляю!!!");
                        document.location.reload();
                    }
                }
            }
        }
    }
}

function drawBall() {
    ctx.beginPath();
    ctx.arc(x, y, ballRadius, 0, Math.PI*2);
    ctx.fillStyle = "#FF0000";
    ctx.fill();
    ctx.closePath();
}

function drawPaddle() {
    ctx.beginPath();
    ctx.rect(paddleX, canvas.height-paddleHeight, paddleWidth, paddleHeight);
    ctx.fillStyle = "#191970";
    ctx.fill();
}
```

```
ctx.closePath();
}
function drawBricks() {
for(var c=0; c<brickColumnCount; c++) {
for(var r=0; r<brickRowCount; r++) {
if(bricks[c][r].status == 1) {
var brickX = (r*(brickWidth+brickPadding))+brickOffsetLeft;
var brickY = (c*(brickHeight+brickPadding))+brickOffsetTop;
bricks[c][r].x = brickX;
bricks[c][r].y = brickY;
ctx.beginPath();
ctx.rect(brickX, brickY, brickWidth, brickHeight);
ctx.fillStyle = "#191970";
ctx.fill();
ctx.closePath();
}
}
}
}
function drawScore() {
ctx.font = "16px Arial";
ctx.fillStyle = "#000000";
ctx.fillText("Счет: "+score, 8, 20);
}
function drawLives() {
ctx.font = "16px Arial";
ctx.fillStyle = "#000000";
ctx.fillText("Жизни: "+lives, canvas.width-65, 20);
}

function draw() {
ctx.clearRect(0, 0, canvas.width, canvas.height);
drawBricks();
drawBall();
drawPaddle();
drawScore();
drawLives();
collisionDetection();

if(x + dx > canvas.width-ballRadius || x + dx < ballRadius) {
dx = -dx;
}
if(y + dy < ballRadius) {
dy = -dy;
}
else if(y + dy > canvas.height-ballRadius) {
if(x > paddleX && x < paddleX + paddleWidth) {
dy = -dy;
}
else {
lives--;
if(!lives) {
alert("Вы проиграли :( ");
document.location.reload();
}
else {
x = canvas.width/2;

```

```
y = canvas.height-30;
dx = 3;
dy = -3;
paddleX = (canvas.width-paddleWidth)/2;
}
}
}

if(rightPressed && paddleX < canvas.width-paddleWidth) {
  paddleX += 7;
}
else if(leftPressed && paddleX > 0) {
  paddleX -= 7;
}

x += dx;
y += dy;
requestAnimationFrame(draw);
}

draw();
```

Рисунок 7 – Код игры

Теперь запустив html-файл выйдет окно с игрой «breakout» написанную пару минут назад(рис.8). Так же ранее было упомянуто о возможности проиграть и выиграть (рис.9-10).

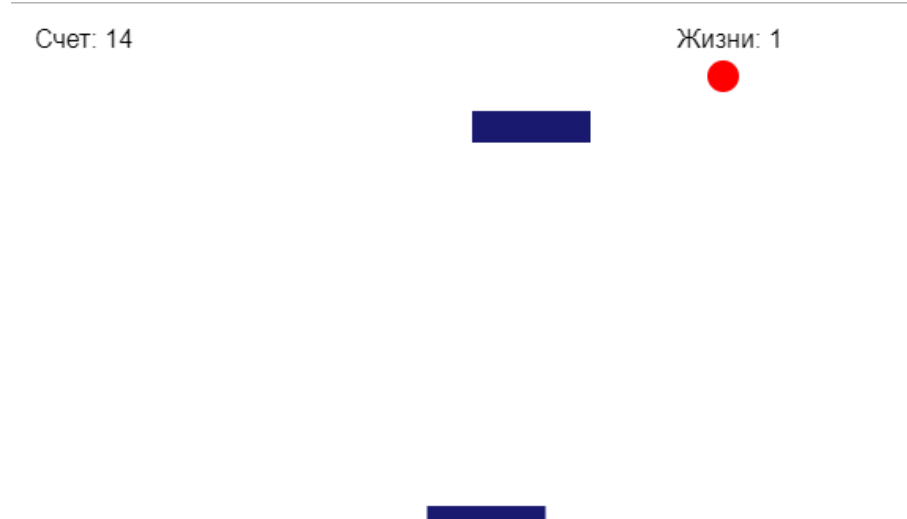


Рисунок 8 – Игра «Breakout»



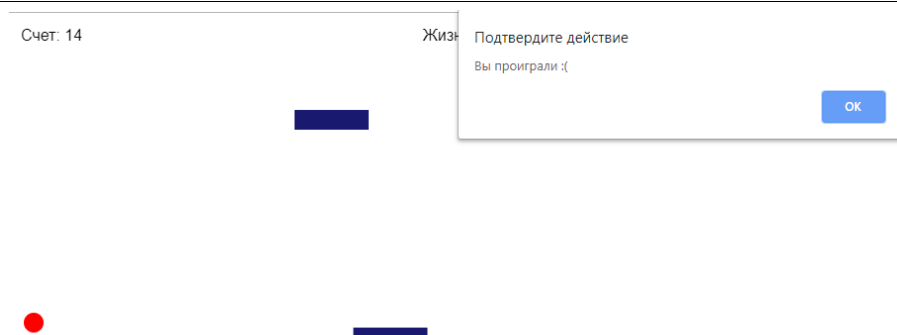


Рисунок 9 – Проигрыш в «breakout»

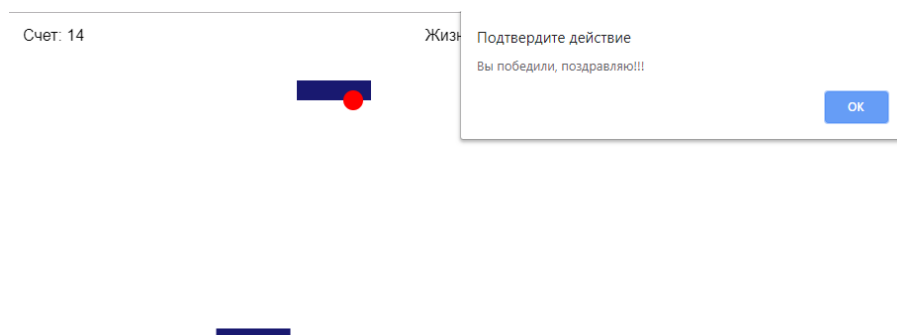


Рисунок 10 – Победа в «breakout»

Таким образом, реализация игры «Breakout» на языке Java возможна и с помощью несложного кода можно поиграть в собственную игру. В результате исследования была произведена пошаговая реализация проекта по разработке браузерной игры «Breakout». Практическим результатом является рабочая браузерная игра.

### Библиографический список

1. Breakout // Википедия URL: <https://ru.wikipedia.org/wiki/Breakout> (дата обращения: 20.08.2019).
2. Беляев С.А. Разработка игр на языке JavaScript: Учебное пособие. СПб.: Лань, 2016. 128с.
3. Захаркина В. В. JavaScript. Основы клиентского программирования: Учебное пособие. СПб.: Ф-т филологии и искусств СПбГУ, 2007. 73 с.
4. Зудилова Т.В., Буркова М.Л Web-программирование JavaScript. СПб: НИУ ИТМО, 2012. 68 с.
5. Что такое JavaScript? Основные функции JavaScript URL: [http://www.progaprosto.ru/doc/yazyk\\_programmirovaniya\\_javascript.php](http://www.progaprosto.ru/doc/yazyk_programmirovaniya_javascript.php)
6. Основы Javascript. URL: <https://learn.javascript.ru/first-steps> (дата обращения 20.08.2018)
7. Прохоренок Н., Дронов В. HTML, JavaScript, PHP и MySQL. СПб.: БХВПетербург, 2015. 766с.
8. Флэнаган Д. JavaScript. Подробное руководство. М.: Символ-Плюс, 2012.

---

357с.

9. Язык программирования JavaScript. URL: [http://progarprosto.ru/doc/yazyk\\_programmirovaniya\\_javascript.php](http://progarprosto.ru/doc/yazyk_programmirovaniya_javascript.php) (дата обращения 20.08.2018).
10. Wiki-учебник по веб-технологиям: JavaScript. URL: <http://www.webmasterwiki.ru/JavaScript> (дата обращения 20.08.2018).
11. Диков А.В. Математические алгоритмы на JavaScript // Известия Пензенского государственного педагогического университета им. В.Г. Белинского. 2009. № 17. С. 84-88.