**Отчёт по лабораторной работе  
Игра на Jquerry**

Для оформления работы в одном стиле, нужно добавить стили. Для работы приложения, следует подключить библиотеки.

<meta charset="UTF-8">

<title>Breakout</title>

<link rel="stylesheet" href="style.css">

<h1 align = "center">Breakout</h1>

<script type="text/javascript">

Страница регистрации будет создана в файле “index.html”. Эта страница будет загружаться первой. Для регистрации необходимо одно поле ввода и кнопка входа.

<html>

<head>

<title>User</title>

<meta http-equiv="Content-Type" content="text/html"; charset="UTF-8" />

</head>

<body>

<h1 align="center">Registration</h1>

<hr />

<br />

<form name="user" method="GET">

<input type="text" placeholder="Name" name="name" ></input>

</form>

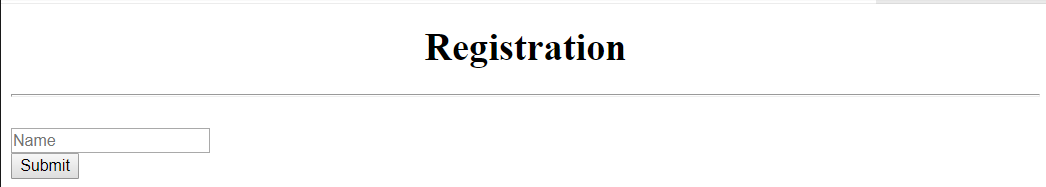
<form id="logbutton" action="game.html">

<button type="submit">Submit</button>

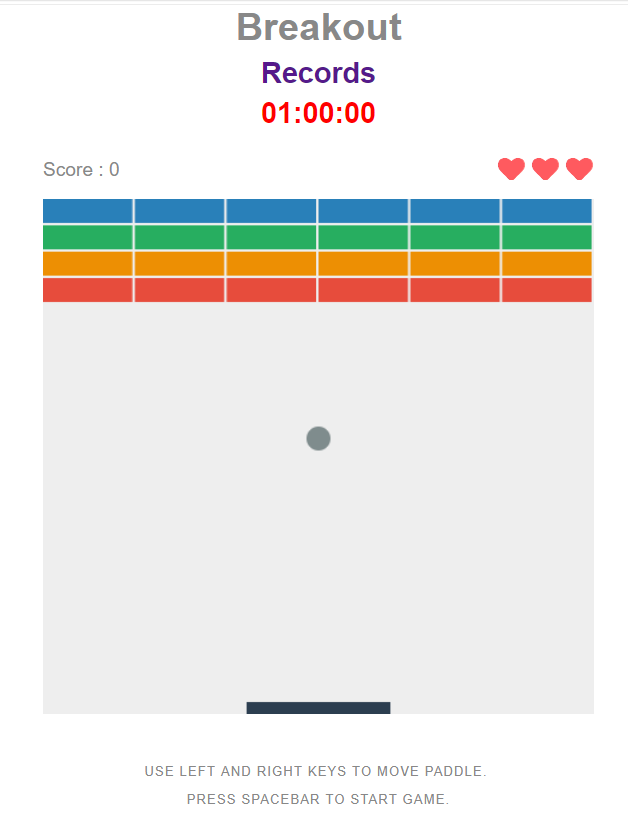
</form>

</body>

</html>

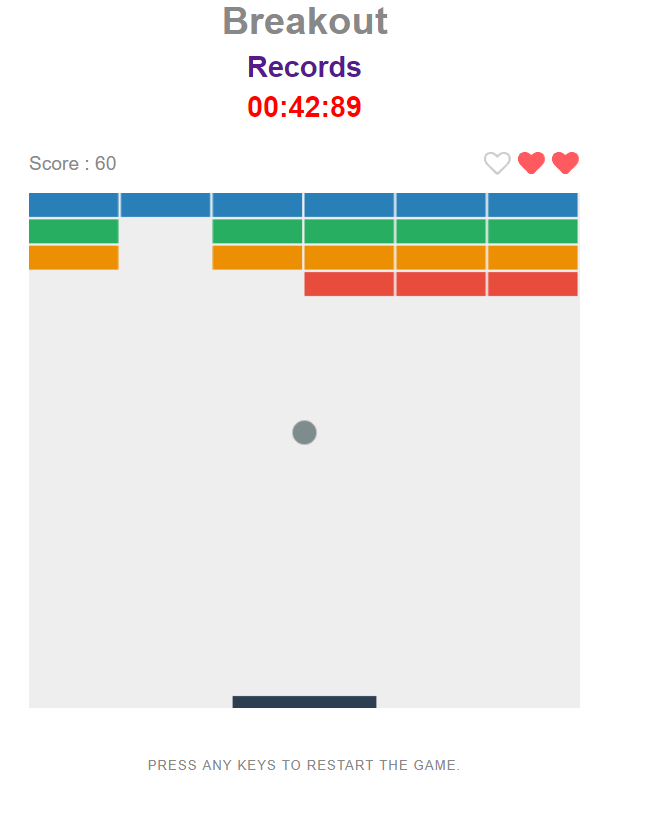


При заполнении поля ввода “name” и нажатии кнопки “submit” мы попадаем на страницу с игрой.



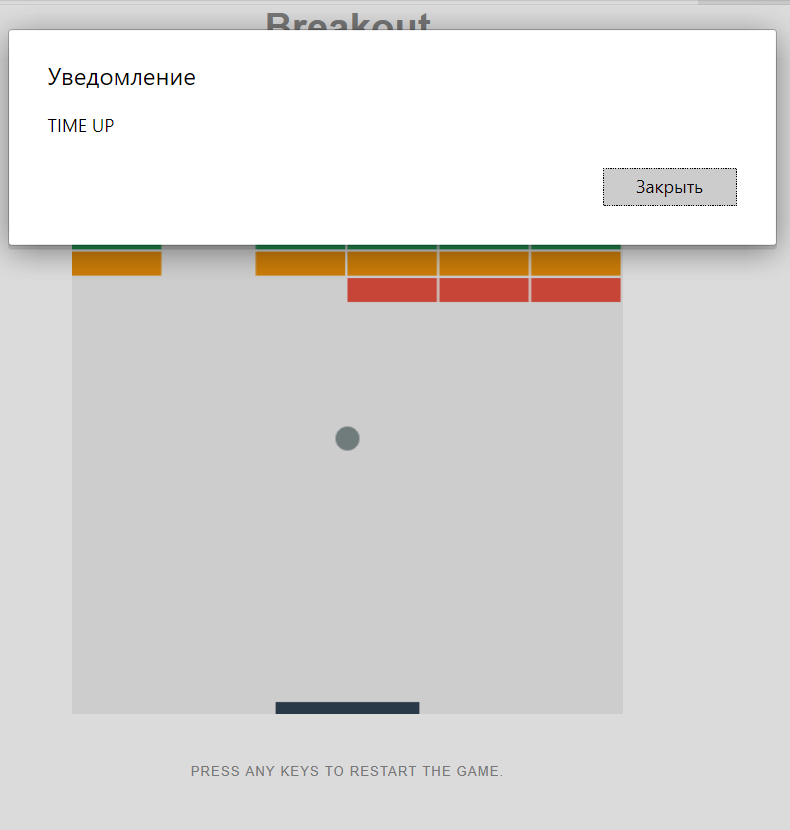
Игра есть арканоид, где нужно, отбивая шарик платформой внизу, разрушить этим же шариком блоки вверху. Управление платформой осуществляется нажатиями клавиш со стрелками влево и вправо для перемещения платформы влево и вправо соответственно.

Для начала игры требуется нажать клавишу “пробел”. Вместе с нажатием клавиши, начнётся обратный отсчёт таймера.



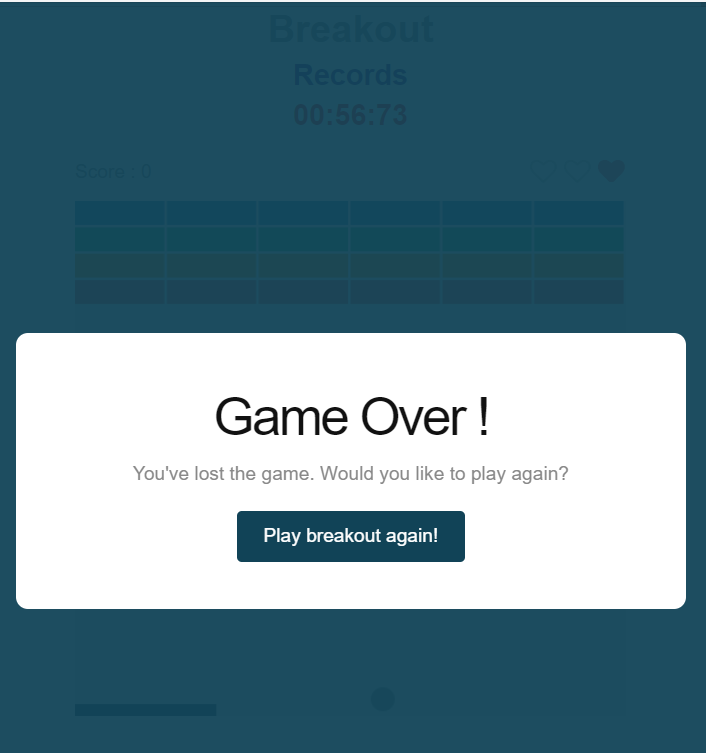
При попадании шарика в нижнюю часть экрана мимо платформы, игрок проигрывает жизнь. У игрока есть 3 жизни. При отнятии жизни, игра переместит шарик в центр поля и приостановит игру, но не таймер, и предложит игроку нажать любую клавишу для продолжения.

Если игрок не успел разбить блоки до истечения таймера, игра выдаст окно с сообщением, на котором есть кнопка “закрыть”. При её нажатии, игра перезапустится.

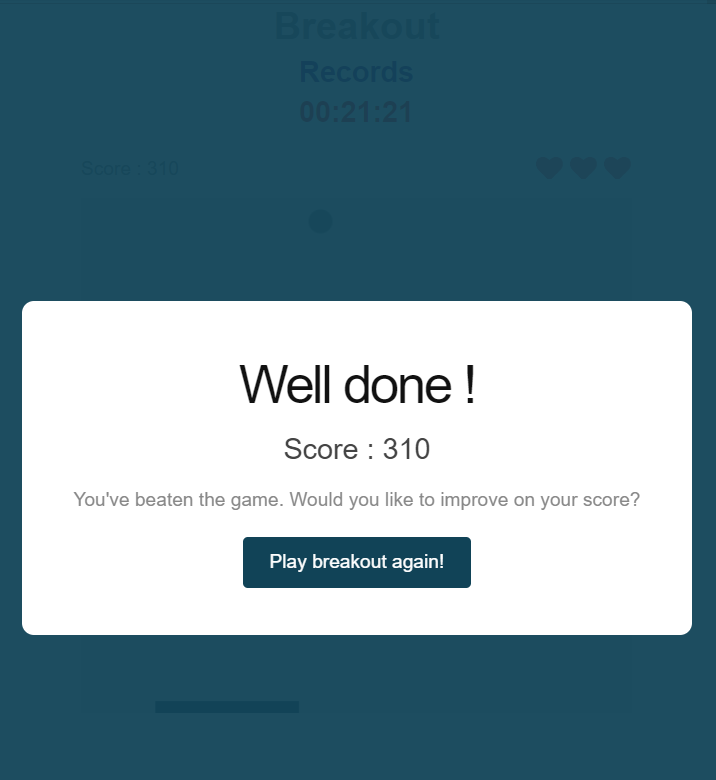


Если игрок проиграет все жизни, выскочит окно с сообщением о проигрыше.

Для перезапуска игры, требуется нажать кнопку “Play breakout again!”.



Если игрок выиграет, выскочит сообщение о победе с финальным счётом игры. Для повторного прохождения игры, требуется нажать кнопку “Play breakout again!”.



Весь игровой процесс умещается в 2 файлах – game.js (сама игра) и game.html (скрипт таймера и оформление страницы).

Game.js

var canvas = $('#breakout-game')[0];

var context = canvas.getContext("2d");

var canvasWidth = canvas.width;

var canvasHeight = canvas.height;

var setting = {

ballColor : "#7f8c8d",

ballRadius : 10,

paddleHeight: 10,

paddleWidth: 120,

frameSpeed : 10,

paddleMoveLeft: false,

paddleMoveRight: false,

paddleMoveRate: 4,

paddleX: 170,

paddleY: 420,

gameState: true,

deathNumber: 0,

deathTotal: 2,

lostMessage: "You've lost the game. Would you like to play again?",

pauseMessage: "Paused. Press spacebar to continue.",

wonMessage: "You've beaten the game. Would you like to improve on your score?",

currentScore: 0,

brickScore: 20,

paddleScore: 10

}

var x = canvasWidth/2, y = 200;

var dx = 0.5, dy = 4;

var bricks = {

row : 4,

column : 6,

color : "#ffc500",

height : 20,

width : undefined,

padding : 2

};

var gameStart, brick, restart,

totalBricks = bricks.row \* bricks.column;

function drawBall() {

context.clearRect(0, 0, canvasWidth, canvasHeight);

context.beginPath();

context.arc(x, y, setting.ballRadius, 0, 2 \* Math.PI, true);

context.closePath();

context.fillStyle = setting.ballColor;

context.fill();

x += dx;

y += dy;

}

function drawPaddle() {

if(setting.paddleMoveLeft && setting.paddleX >= 0) {

setting.paddleX -= setting.paddleMoveRate;

}

if(setting.paddleMoveRight && setting.paddleX <= canvasWidth - setting.paddleWidth) {

setting.paddleX += setting.paddleMoveRate;

}

context.fillStyle = '#2c3e50';

context.fillRect( setting.paddleX, setting.paddleY, setting.paddleWidth, setting.paddleHeight);

}

function movePaddle() {

$(document).on('keydown', function(e){

if( e.keyCode == 39 ) {

setting.paddleMoveRight = true;

} else if( e.keyCode == 37 ) {

setting.paddleMoveLeft = true;

}

}).on('keyup', function(e){

if( e.keyCode == 39 ) {

setting.paddleMoveRight = false;

} else if( e.keyCode == 37 ) {

setting.paddleMoveLeft = false;

}

});

}

function initBricks() {

brick = new Array(bricks.column);

for( i = 0; i < bricks.column; i++ ) {

brick[i] = new Array(bricks.row);

for( j = 0; j < bricks.row; j++ ) {

brick[i][j] = { x : 0, y : 0, state : true };

}

}

}

function drawBricks() {

bricks.width = ( (canvasWidth) / bricks.column) - 2;

for( i = 0; i < bricks.column; i++ ) {

for( j = 0; j < bricks.row; j++ ) {

if ( brick[i][j].state ) {

var brickX = i \* (bricks.width + bricks.padding);

var brickY = j \* (bricks.height + bricks.padding);

brick[i][j].x = brickX;

brick[i][j].y = brickY;

context.beginPath();

context.rect( brickX, brickY, bricks.width, bricks.height);

if( j == 0) {

context.fillStyle = '#2980b9';

} else if( j == 1 ) {

context.fillStyle = '#27ae60';

} else if( j == 2 ) {

context.fillStyle = '#ED8F03';

} else {

context.fillStyle = '#e74c3c';

}

context.fill();

context.closePath();

}

}

}

}

function brickCollision() {

for( n = 0; n < bricks.column; n++ ) {

for( k = 0; k < bricks.row; k++ ) {

var currentBrick = brick[n][k];

if(currentBrick.state) {

if( x > currentBrick.x && x < currentBrick.x + bricks.width && y > currentBrick.y && y < currentBrick.y + bricks.height ) {

dy = -dy;

currentBrick.state = false;

setting.currentScore += setting.brickScore;

$('.instr').text('Score : ' + setting.currentScore);

totalBricks -= 1;

}

}

}

}

}

function boundaryCollision() {

if( y - setting.ballRadius <= 0) {

dy = -dy;

}

else if ( y + setting.ballRadius >= canvasHeight ) {

var fullBallLeft = x + setting.ballRadius;

var fullBallRight = x - setting.ballRadius;

if( fullBallLeft >= setting.paddleX && fullBallRight < setting.paddleX + setting.paddleWidth) {

dx = 8 \* ( ( x - (setting.paddleX + setting.paddleWidth/2) ) / setting.paddleWidth);

dy = Math.sqrt( 20 - ( dx \* dx) );

dy = -dy;

if( setting.currentScore > 0 ) {

setting.currentScore -= setting.paddleScore;

}

$('.instr').text('Score : ' + setting.currentScore);

}

else {

clearInterval(gameStart);

setting.gameState = false;

}

}

if( y + setting.ballRadius > canvasHeight + 15) {

clearInterval(gameStart);

setting.gameState = false;

}

if( x + setting.ballRadius >= canvasWidth || x - setting.ballRadius <= 0) {

dx = -dx;

}

}

function loseTurnCallback() {

$('.life .heart').eq(setting.deathNumber).addClass('heart-o');

setTimeout(function(e){

$('.instr-footer small').text('Press any keys to restart the game.');

setting.deathNumber += 1;

x = canvasWidth/2;

y = 200;

setting.paddleX = 170;

drawBall();

drawBricks();

drawPaddle();

setting.gameState = true;

$(document).on('keyup', function(e) {

gameStart = setInterval( init , setting.frameSpeed);

$(this).off('keyup');

$('.instr-footer small').text('');

});

}, 400);

}

function loseGameCallback() {

modalPopup(setting.lostMessage, "Game Over !");

startTimer(false);

}

function initSpacebar() {

$(document).on('keyup', function(e) {

if( e.keyCode == 32 && setting.deathNumber < setting.deathTotal ) {

gameStart = setInterval( init , setting.frameSpeed);

startTimer(true);

$('.instr-footer small').text('');

// Only allows one-time click on document load

$(this).off('keyup');

}

});

}

function gameWon() {

clearInterval(gameStart);

$('.modal-box h3').after("<p class='modal-score'>Score : " + setting.currentScore + "</p>");

modalPopup(setting.wonMessage, "Well done !");

drawBall();

drawPaddle();

}

function modalPopup(message, header) {

$('.modal-message').text(message);

$('.modal-box h3').text(header);

setTimeout(function(){

$('.modal-overlay').show();

$('body').addClass('game-over');

}, 100);

$('.btn-play-again').on('click', function(e){

e.preventDefault();

location.reload();

});

}

function init() {

if( setting.gameState == true ) {

drawBall();

}

boundaryCollision()

drawPaddle();

drawBricks();

movePaddle();

brickCollision();

if( setting.gameState == false && setting.deathNumber < setting.deathTotal ) {

loseTurnCallback();

} else if( setting.gameState == false && setting.deathNumber == setting.deathTotal ) {

loseGameCallback();

}

if( totalBricks == 0 ) {

gameWon();

startTimer(false);

}

}

initBricks();

init();

initSpacebar();

Game.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Breakout</title>

<link rel="stylesheet" href="style.css">

<h1 align = "center">Breakout</h1>

<script type="text/javascript">

function startTimer(bo) {

var my\_timer = document.getElementById("my\_timer");

var time = my\_timer.innerHTML;

var arr = time.split(":");

var h = arr[0];

var m = arr[1];

var s = arr[2];

if (bo) boo=true;

if (bo==false) boo=false;

if (s == 0) {

if (m == 0) {

if (h == 0) {

alert("TIME UP");

window.location.reload();

return;

}

h--;

m = 60;

if (h < 10) h = "0" + h;

}

m--;

if (m < 10) m = "0" + m;

s = 99;

}

else s--;

if (s < 10) s = "0" + s;

document.getElementById("my\_timer").innerHTML = h+":"+m+":"+s;

if (boo) {setTimeout(startTimer, 10);}

else if (boo==false) {setTimeout(startTimer, 100000000000000);}

}

</script>

<h2 align = "center"><a href="records.html">Records</a></h2>

</head>

<body>

<p align = "center"><span id="my\_timer" style="color: #f00; font-size: 150%; font-weight: bold;" >01:00:00</span></p>

<!-- Top Bar -->

<div class="game-notifications clearfix">

<div class="instr">

Score : 0

</div>

<ul class="life">

<li class="heart"></li>

<li class="heart"></li>

<li class="heart"></li>

</ul>

</div>

<!-- Game Container -->

<div class="container">

<canvas id="breakout-game" height="430" width="460"></canvas>

</div>

<div class="container">

<div class="instr-footer">

<small>

Use left and right keys to move paddle.

<br>

Press spacebar to start game.

</small>

</div>

</div>

<!-- Modal for game over -->

<div class="modal-overlay">

</div>

<div class="modal-box">

<h3>Game Over</h3>

<p class="modal-message">Congratulations, you've beaten the game. Would you like to play again?</p>

<a href="#" class="btn-play-again">Play breakout again!</a>

</div>

<!-- Load Google Font Roboto -->

<script type="text/javascript">

WebFontConfig = {

google: { families: [ 'Roboto::latin' ] }

};

(function() {

var wf = document.createElement('script');

wf.src = ('https:' == document.location.protocol ? 'https' : 'http') +

'://ajax.googleapis.com/ajax/libs/webfont/1/webfont.js';

wf.type = 'text/javascript';

wf.async = 'true';

var s = document.getElementsByTagName('script')[0];

s.parentNode.insertBefore(wf, s);

})();

</script>

<!-- Load Jquery 1.11.3 -->

<script src="jquery.js"></script>

<script src="game.js"></script>

</body>

</html>

Также, в верхней части поля, есть кнопка “Records”, нажав на которую, мы попадаем на страницу с рекордами.

