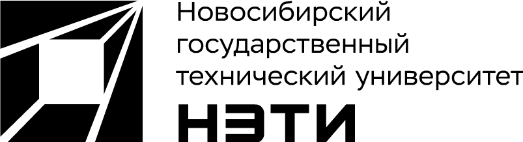
**ОТЧЕТ  
ПО ЛАБОРАТОРНОЙ РАБОТЕ №4  
ПО ДИСЦИПЛИНЕ «WEB-ПРОГРАММИРОВАНИЕ»**

**Язык программирования JavaScript  
  
  
**

Факультет: АВТ Преподаватель:

Группа: АВТ-813 [Новицкая Юлия Вадимовна](https://dispace.edu.nstu.ru/personal/index/5186)

Студент(ы):

Лысак Кирилл

**Оглавление:**

[**Цель работы: 3**](#_Toc74607529)

[**Задание: 3**](#_Toc74607530)

[**Вариант 9: 3**](#_Toc74607531)

[**Выполнение: 4**](#_Toc74607532)

[**Выводы: 9**](#_Toc74607533)

[**Приложение A: 10**](#_Toc74607534)

[***index.html* 10**](#_Toc74607535)

[***styles.css* 10**](#_Toc74607536)

[***main.js* 12**](#_Toc74607537)

****Цель работы:****

Изучить основы клиентского скриптового языка JavaScript, работу с объектной моделью документа DOM (Document Object Model), познакомиться с возможностями, предоставляемые фреймворком jQuery.

****Задание:****

Создать клиентский скрипт на языке JavaScript, выполняющий действия в соответствии с вариантом. Использовать возможности, предоставляемые объектной моделью документа DOM, использовать фреймворк jQuery (или аналогичный).

****Вариант 9:****

Игра «Падающие мячи». По игровому полю сверху вниз в случайном порядке падают мячи, которые нужно ловить корзиной, передвигаемой горизонтально вдоль нижней границы игрового поля. Игру можно начать и прекратить щелчком по соответствующей кнопке. Со временем скорость падения мячей может увеличиваться. После остановки игры выводится результат — процент пойманных мячей. Среди прочего использовать возможности, предоставляемые фреймворком jQuery.

****Выполнение:****

В главном меню присутствует 3 кнопки (*Рисунок 1*):

«AGAIN» – запускает игру и позволяет начать её снова;

«PAUSE» – Работает только после включения игры и ставит игру на паузу;

«EXIT» – Осуществляет выход в главное меню;

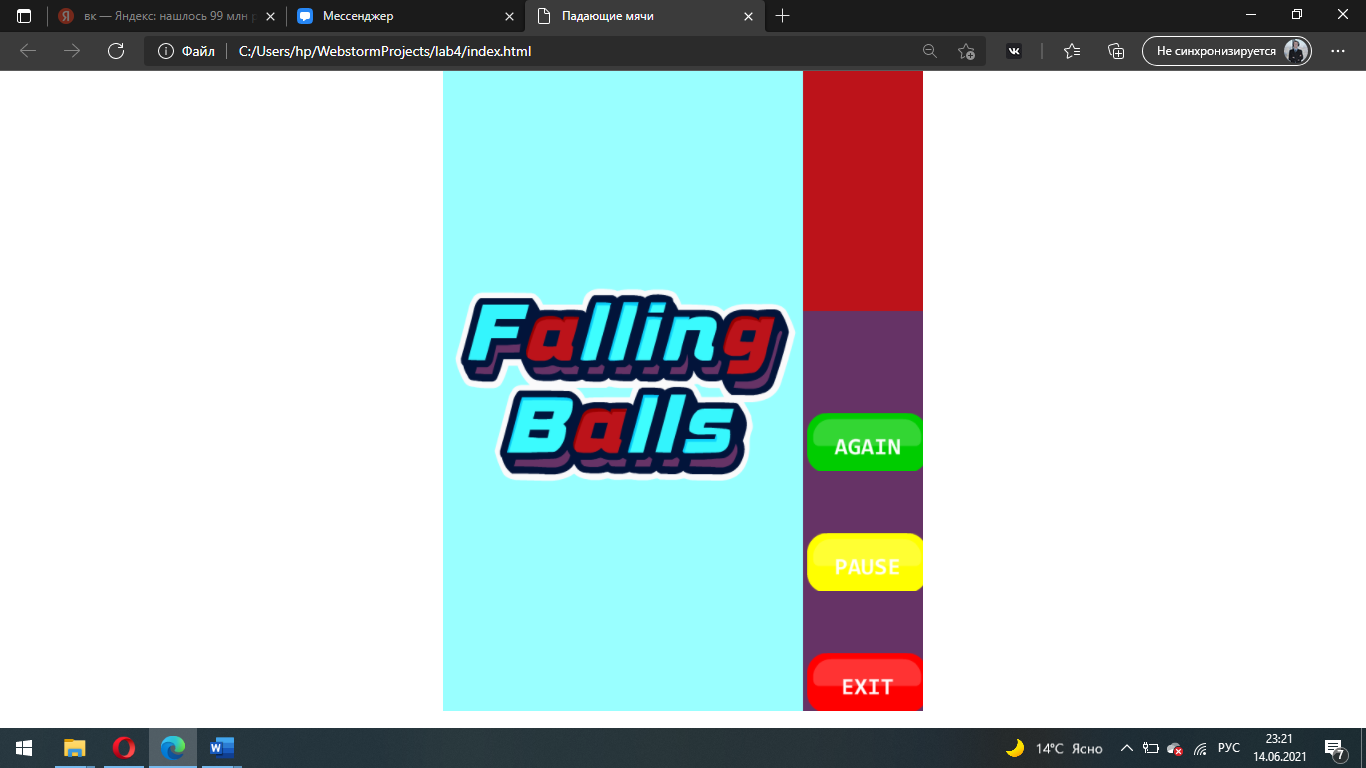


Рисунок 1 – Главное меню программы

После запуска игры на экране появляется платформа, движение которой осуществляется с помощью стрелок на клавиатуре, а также начинают падать шары сверху игрового поля. Задачи игрока поймать как можно больше шаров (*Рисунок 2*). Данная игра является аркадой и не имеет конца.



Рисунок 2 – Запуск игры

На боковой панели справа для игрока отображается счётчик пойманных шаров, количество жизней, время, проведённое игроком с начала новой игры, а также номер уровня. Новый уровень достигается при прохождении интервала времени 10 секунд. Он сопровождается плавным появлением надписи на поле игры «New level», которая была осуществлена с помощью jQuery и свойства opacity. С новым уровнем в игре меняется цвет карты, цвет падающих шаров. Чтобы уровень отличался от предыдущего не только визуально, количество падающих шаров увеличивается, как и их скорость. Максимальный уровень в игре девятый (*Рисунок 3 - 4*).

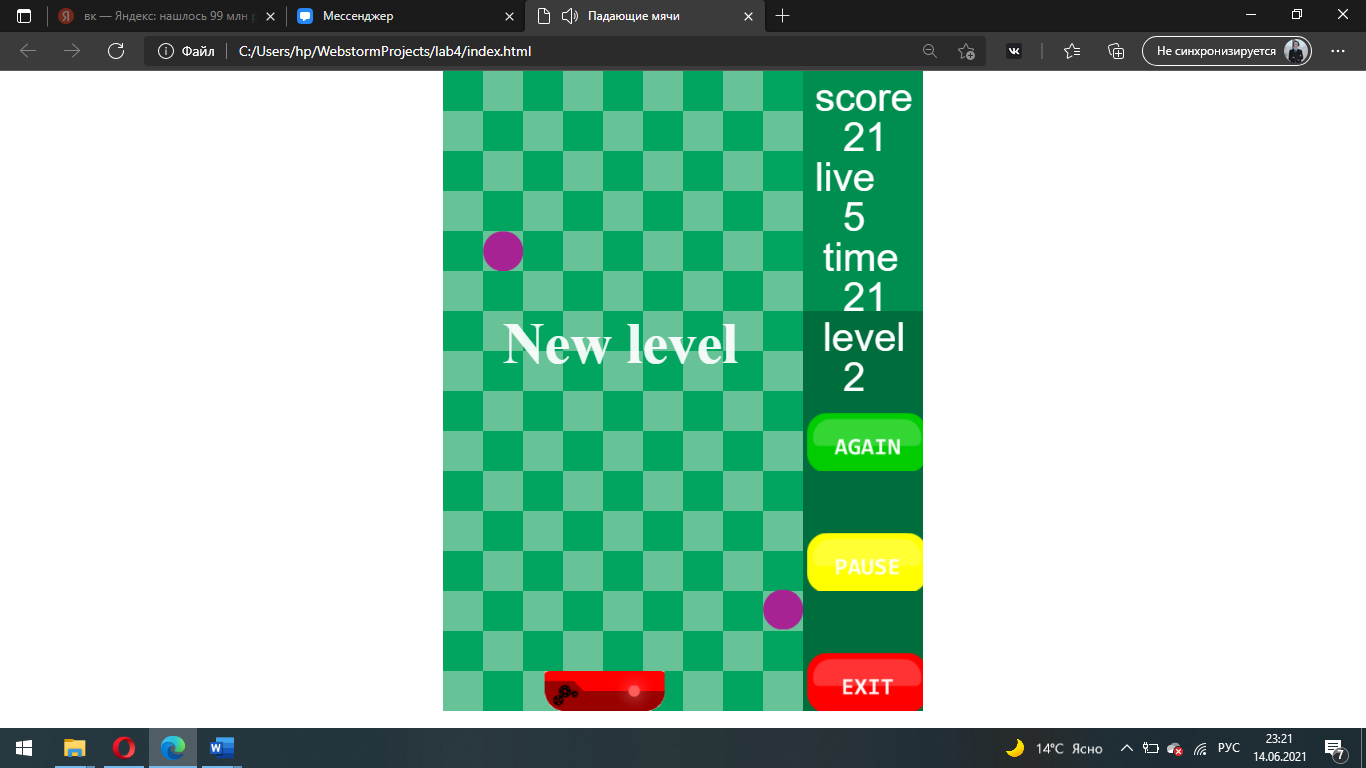


Рисунок 3 – Новый уровень

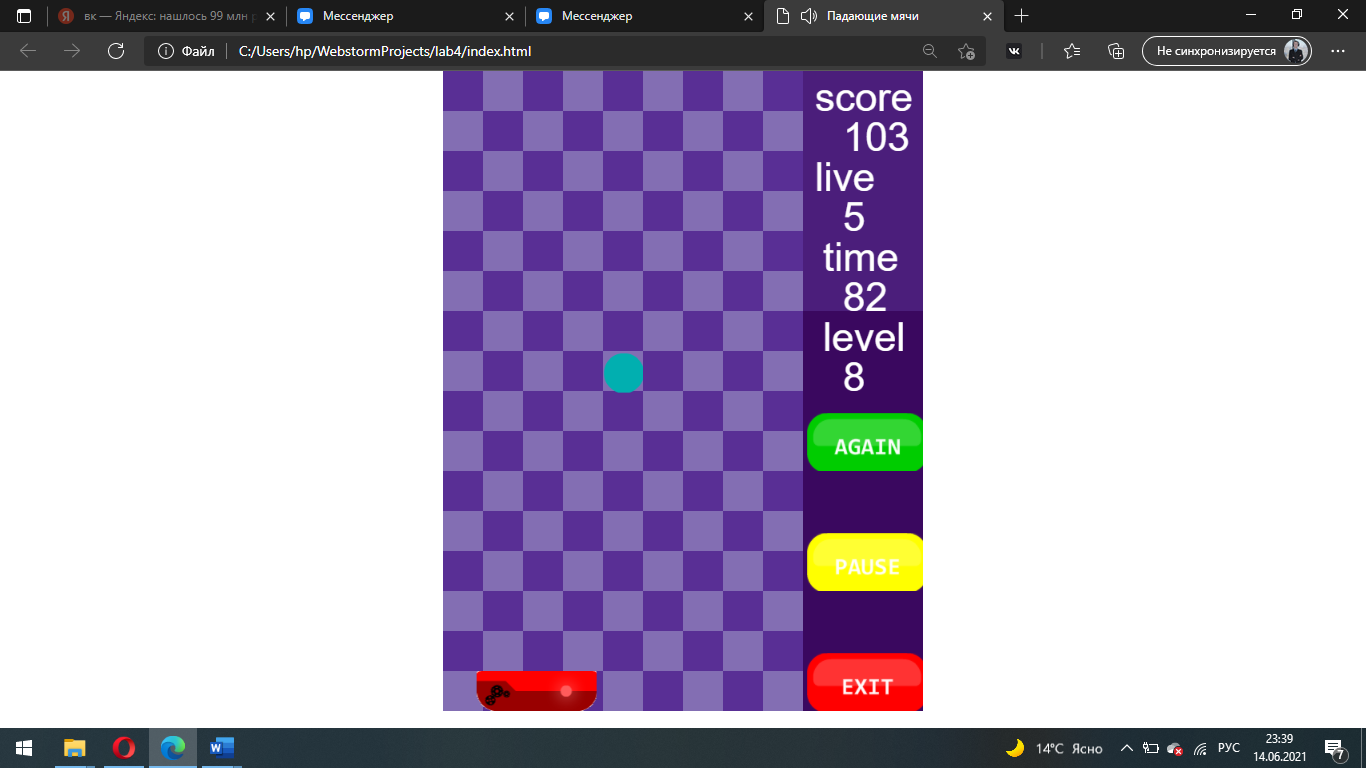


Рисунок 4 – Новый уровень

По окончанию количества жизней у игрока появляется табличка с концом игры (*Рисунок 5*).

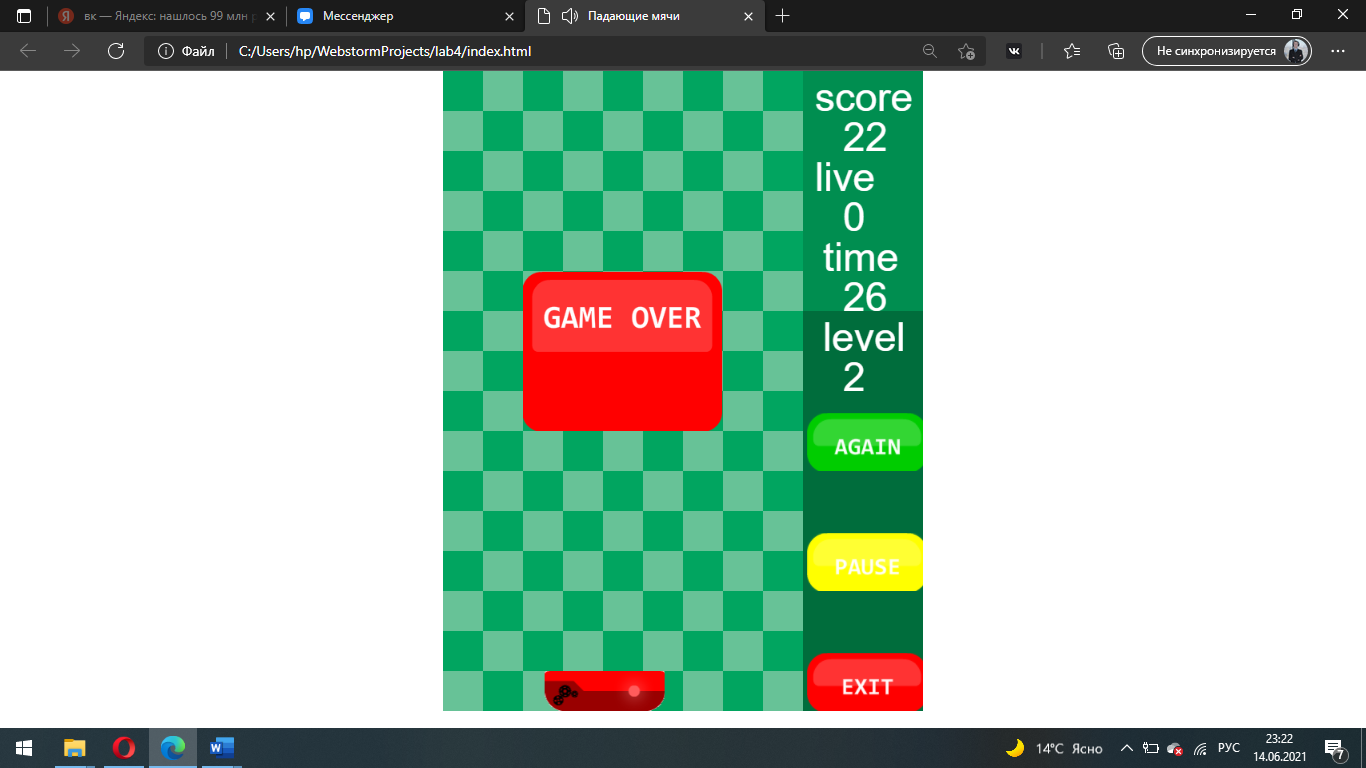


Рисунок 5 – Конец игры

На данном рисунке (*Рисунок 6*) можно увидеть все текстуры, использованные в игре. Они были нарисованы в векторном редакторе Adobe Animate.



Рисунок 6 – Текстуры

Также в игру была добавлена музыка и звуки при достижении нового уровня, пойманном шаре и проигрыше (*Рисунок 7*).

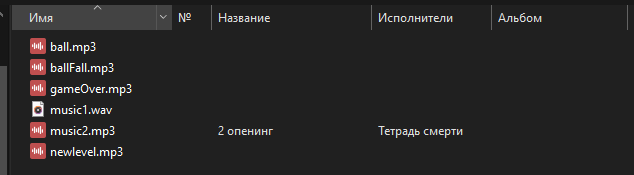


Рисунок 7 – Аудио

****Выводы:****

Мы изучили основы клиентского скриптового языка JavaScript, работу с объектной моделью документа DOM (Document Object Model), познакомились с возможностями, предоставленными фреймворком jQuery.

****Приложение A:****

*****index.html*****

<!DOCTYPE html>  
<html lang="ru">  
<head>  
 <meta charset="UTF-8">  
 <title>Падающие мячи</title>  
 <link rel="stylesheet" href="styles.css">  
 <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>  
  
  
  
</head>  
<body>  
  
<div class="game\_space">  
 <!--<h1 id ="Level">Level</h1>-->  
 <canvas id = "FallingBalls" width="600" height="800"></canvas>  
 <button id="btn1" class="button1" onclick="StopGame()"> </button>  
 <button id="btn2" class="button2" onclick=PauseGame()> </button>  
 <button id="btn3" class="button3" onclick=StartGame()> </button>  
 <div id = "text"></div>  
</div>  
  
<script src = "main2.js"></script>  
<script>  
  
  
</script>  
</body>  
</html>

*****styles.css*****

canvas {  
 /\* Распалагаем окно игры по центру \*/  
 background: url(files/images/ground0.png);  
 display: block;  
 margin: 0 auto;  
}  
  
body, html  
{  
 width: 100%;  
 height: 100%;  
 padding: 0px;  
 margin: 0px;  
 overflow: hidden;  
}  
/\*#Level  
{  
 width: 200px;  
 background: #fc0;  
 padding: 5px;  
 border: solid 1px black;  
 float: left;  
 position: relative;  
 top: 0px;  
 left: 0px;  
}\*/  
  
  
.button1  
{  
 width:145px;  
 height:73px;  
 background: url(files/images/buttonImg1.png);  
 position:absolute;  
 bottom: 0px; /\*Положение от нижнего края \*/  
 right: 0px; /\*Положение от правого края \*/  
 color: #fff;  
 border: 0px solid #fff;  
 text-align: center;  
 outline: none;  
 text-decoration: none;  
 cursor: pointer;  
  
}  
.button2  
{  
 width:145px;  
 height:73px;  
 background: url(files/images/buttonImg2.png);  
 position:absolute;  
 bottom: 150px; /\*Положение от нижнего края \*/  
 right: 0px; /\*Положение от правого края \*/  
 color: #fff;  
 border: 0px solid #fff;  
 text-align: center;  
 cursor: pointer;  
 outline: none;  
  
 text-decoration: none;  
}  
.button3  
{  
 width:145px;  
 height:73px;  
 background: url(files/images/buttonImg3.png);  
 position:absolute;  
 bottom: 300px; /\*Положение от нижнего края \*/  
 right: 0px; /\*Положение от правого края \*/  
 color: #fff;  
 border: 0px solid #fff;  
 text-align: center;  
 outline: none;  
 cursor: pointer;  
 text-decoration: none;  
}  
  
#text{  
 font-weight: 800;  
 color: white;  
 font-family: 'Times New Roman', Times, serif; /\* Гарнитура текста \*/  
 font-size: 450%; /\* Размер шрифта в процентах \*/  
 position:absolute;  
 left: 75px; /\*Положение от нижнего края \*/  
 top: 300px; /\*Положение от правого края \*/  
}  
  
.game\_space {  
 width:600px;  
 height:800px;  
 position:relative;  
 display: block;  
 margin: 0 auto;  
}  
button:active, button:focus {  
 outline: none;  
}  
button::-moz-focus-inner {  
 border: 0;  
}

*****main.js*****

const canvas = document.getElementById("FallingBalls");  
const ctx = canvas.getContext("2d");  
  
window.addEventListener("keydown", function (e) { KeyDown(e); }); //Получение нажатий с клавиатуры  
const ground = new Image();  
ground.src = "files/images/ground.png";  
const gameOver = new Image();  
gameOver.src = "files/images/gameOver.png";  
  
  
let box = 50; /\* универсаньный размер нашей одной клетки поля\*/  
var a = 1;  
let score; /\* Счётчик результата\*/  
let level = 0;  
let live;  
let speedBasket;  
let speedBall;  
let speedSpawnBall;  
var player;  
let time = 0; /\* Счётчик времени\*/  
var sost;  
var game\_time; // таймер игрового времени  
var game\_run = false;  
var isResizeble = false;  
  
  
ballImage = new Array();  
ballImage[0] = "files/images/ballImg1.png"  
ballImage[1] = "files/images/ballImg2.png"  
ballImage[2] = "files/images/ballImg3.png"  
ballImage[3] = "files/images/ballImg4.png"  
ballImage[4] = "files/images/ballImg5.png"  
ballImage[5] = "files/images/ballImg6.png"  
ballImage[6] = "files/images/ballImg7.png"  
ballImage[7] = "files/images/ballImg8.png"  
  
var sound1 = new Audio('files/sounds/ball.mp3');  
var sound2 = new Audio('files/sounds/newLevel.mp3');  
var sound3 = new Audio('files/sounds/music1.wav');  
var sound4 = new Audio('files/sounds/music2.mp3');  
var sound5 = new Audio('files/sounds/ballFall.mp3');  
var sound6 = new Audio('files/sounds/gameOver.mp3');  
  
function soundClick()  
{  
  
 sound1.play();  
  
}  
function soundClick2()  
{  
  
 sound2.play();  
  
}  
function soundClick3()  
{  
  
  
  
}  
  
function StopGame()  
{  
 game\_run = !game\_run;  
 Stop();  
}  
function PauseGame()  
{  
 game\_run = !game\_run;  
 if (game\_run)  
 {  
 Start();  
 game\_time = setInterval(tick, 1000);  
  
  
 }  
 else  
 {  
 Stop();  
 }  
}  
function StartGame()  
{  
 if( time == 0)  
 {  
 Start();  
 init();  
 }  
 if( game\_run)  
 {  
  
 Stop();  
 Start();  
 init();  
  
 }  
 else  
 {  
 if( time != 0)  
 {  
 Stop();  
 }  
 }  
 score = 0; /\* Счётчик результата\*/  
 level = 0;  
 live = 5;  
 speedBasket = 30;  
 speedBall = 10;  
 speedSpawnBall = 1;  
 player = 0;  
 time = 0; /\* Счётчик времени\*/  
 sost = true;  
  
 backgrounds = //Массив с фонами  
 [  
 new Backgrounds("files/images/ground.png", 0),  
 new Backgrounds("files/images/ground2.png", 0),  
 new Backgrounds("files/images/ground3.png", 0),  
 new Backgrounds("files/images/ground4.png", 0),  
 new Backgrounds("files/images/ground5.png", 0),  
 new Backgrounds("files/images/ground6.png", 0),  
 new Backgrounds("files/images/ground7.png", 0),  
  
 ];  
 objects = //Массив игровых объектов  
 [  
 new Basket("files/images/basketImg.gif", 3 \* box, 15 \* box),  
 ];  
 objects2 = //Массив игровых объектов  
 [  
   
 ];  
  
  
 game\_run = !game\_run;  
  
  
}  
  
class Ball  
{  
 constructor(image, x, y)  
 {  
 this.xBall = x;  
 this.yBall = y;  
  
 this.imageBall= new Image();  
 this.dead = false;  
 this.imageBall.src = image;  
 }  
 Update()  
 {  
 if(this.yBall > 800)  
 {  
 live--;  
 sound5.play();  
 this.dead = true;  
 }  
 else  
 {  
 this.yBall += speedBall;  
 }  
 }  
  
}  
var objects2 = //Массив игровых объектов  
 [  
  
 ];  
  
  
var SpawnBall = function()  
{  
 /\*alert("fff");\*/  
 clearInterval(interval);  
 if ((time > 1) && (time < 10)) /\* Изменяем скорость в зависимости от времени\*/  
 {  
 a=Math.round(Math.random()\*3+1)  
  
 speedBall = 7.5;  
 speedSpawnBall = 1;  
 speedBasket = 30;  
  
 }  
 if ((time > 10) && (time < 20)) /\* Изменяем скорость в зависимости от времени\*/  
 {  
  
  
 a=Math.round(Math.random()\*3+4)  
 backgrounds[0].Update(backgrounds[1]);  
 speedBall = 8;  
 speedSpawnBall = 1.1;  
 speedBasket = 37.5;  
  
 }  
 if ((time > 30) && (time < 40)) /\* Изменяем скорость в зависимости от времени\*/  
 {  
 a=Math.round(Math.random()\*3+7)  
 backgrounds[0].Update(backgrounds[1]);  
 speedBall = 8.5;  
 speedSpawnBall = 1.2;  
 speedBasket = 40;  
 }  
 if ((time > 40) && (time < 50)) /\* Изменяем скорость в зависимости от времени\*/  
 {  
 a=Math.round(Math.random()\*3+10)  
 backgrounds[0].Update(backgrounds[2]);  
 speedBall = 9;  
 speedSpawnBall = 1.3;  
 speedBasket = 45;  
 }  
 if ((time > 60) && (time < 70)) /\* Изменяем скорость в зависимости от времени\*/  
 {  
 a=Math.round(Math.random()\*3+13)  
 backgrounds[0].Update(backgrounds[3]);  
 speedBall = 10;  
 speedSpawnBall = 1.5;  
 speedBasket = 50;  
 }  
 if ((time > 80) && (time < 90)) /\* Изменяем скорость в зависимости от времени\*/  
 {  
 a=Math.round(Math.random()\*3+16)  
 backgrounds[0].Update(backgrounds[4]);  
 speedBall = 11;  
 speedSpawnBall = 1.7;  
 speedBasket = 60;  
 }  
 if ((time > 90) && (time < 100)) /\* Изменяем скорость в зависимости от времени\*/  
 {  
 sound3.volume=0;  
 sound4.play();  
 sound4.volume=0.2;  
 a=Math.round(Math.random()\*3+19)  
 backgrounds[0].Update(backgrounds[5]);  
 speedBall = 12.5;  
 speedSpawnBall = 2;  
 speedBasket = 75;  
 }  
 if ((time > 90) && (time < 100)) /\* Изменяем скорость в зависимости от времени\*/  
 {  
 a=Math.round(Math.random()\*3+22)  
 backgrounds[0].Update(backgrounds[6]);  
 speedBall = 13;  
 speedSpawnBall = 2.5;  
 speedBasket = 90;  
 }  
 /\*var i = (Math.floor(Math.random() \* 1 + 0));\*/  
 objects2.push( new Ball("files/images/ballImg"+a+".png", (Math.floor(Math.random() \* 9 + 0) \* box),0));  
  
 interval = setInterval(SpawnBall, 1000 / speedSpawnBall);  
}  
var interval = setInterval(SpawnBall, 1000 / speedSpawnBall);  
  
  
  
function Stop()  
{  
  
  
 clearInterval(timer); //Остановка обновления  
 clearInterval(interval); //Остановка обновления  
 clearInterval(game\_time); //Остановка обновления  
}  
  
function KeyDown(e)  
{  
 switch(e.keyCode) {  
 case 37: //Влево  
 if (objects[player].xBasket > 0)  
 {  
 objects[player].Move("x", -speedBasket);  
 }  
 break;  
  
 case 39: //Вправо  
 if(objects[player].xBasket < 300)  
 {  
 objects[player].Move("x", speedBasket);  
 }  
 break;  
  
 case 38: //Вверх  
 break;  
  
 case 40: //Вниз  
 break;  
  
 case 27: //Esc  
 break;  
 }  
}  
  
class Backgrounds  
{  
 constructor(image, y)  
 {  
 this.xBackground = 0;  
 this.yBackground = y;  
  
 this.imageBackground = new Image();  
  
 this.imageBackground.src = image;  
 }  
 Update(background)  
 {  
 this.imageBackground = background.imageBackground;  
 }  
  
}  
  
  
class Basket  
{  
 constructor(image, x, y)  
 {  
 this.xBasket = x;  
 this.yBasket = y;  
 this.widthBasket = 3 \* box;  
 this.heightBasket = box;  
  
 this.imageBasket = new Image();  
  
 this.imageBasket.src = image;  
 }  
  
 Update()  
 {  
  
 }  
  
 Move(v, dx)  
 {  
  
 if(v == "x") //Перемещение по оси X  
 {  
 this.xBasket += dx; //Смещение  
 event.clientX  
 }  
  
 }  
 Collide(ball)  
 {  
 var hit = false;  
  
 if(this.yBasket < ball.yBall + ball.imageBall.height && this.yBasket + this.imageBasket.height > ball.yBall) //Если объекты находятся на одной линии по горизонтали  
 {  
 if(this.xBasket + this.imageBasket.width > ball.xBall && this.xBasket < ball.xBall + ball.imageBall.width) //Если объекты находятся на одной линии по вертикали  
 {  
 soundClick();  
 ball.dead = true;  
 hit = true;  
 }  
 }  
  
 return hit;  
 }  
  
}  
var MouseCoords =  
 {  
  
 // X-координата  
 getX: function(e)  
 {  
 if (e.pageX)  
 {  
 return e.pageX;  
 }  
 else if (e.clientX)  
 {  
 return e.clientX+(document.documentElement.scrollLeft || document.body.scrollLeft) - document.documentElement.clientLeft;  
 }  
  
 return 0;  
 },  
  
 // Y-координата  
 getY: function(e)  
 {  
 if (e.pageY)  
 {  
 return e.pageY;  
 }  
 else if (e.clientY)  
 {  
 return e.clientY+(document.documentElement.scrollTop || document.body.scrollTop) - document.documentElement.clientTop;  
 }  
  
 return 0;  
 }  
}  
  
  
function init() /\* функция запуска секундомера\*/  
{  
 time = 0;  
 game\_time = setInterval(tick, 1000);  
}  
  
function tick() /\* Секундомер\*/  
{  
 if(time == 0)  
 {  
 soundClick2();  
 $("#text").fadeOut(1);  
 }  
 $("#text").text("New level");  
 if(time == 1)  
 {  
 soundClick2();  
 $("#text").fadeIn(500);  
 $("#text").fadeOut(500);  
 }  
 if(time == 10)  
 {  
 soundClick2();  
 level++;  
 $("#text").fadeIn(500);  
 $("#text").fadeOut(500);  
 }  
 if(time == 20)  
 {  
 soundClick2();  
 level++;  
 $("#text").fadeIn(500);  
 $("#text").fadeOut(500);  
 }  
 if(time == 30)  
 {  
 soundClick2();  
 level++;  
 $("#text").fadeIn(500);  
 $("#text").fadeOut(500);  
 }  
 if(time == 40)  
 {  
 soundClick2();  
 level++;  
 $("#text").fadeIn(500);  
 $("#text").fadeOut(500);  
 }  
 if(time == 50)  
 {  
 soundClick2();  
 level++;  
 $("#text").fadeIn(500);  
 $("#text").fadeOut(500);  
 }  
 if(time == 60)  
 {  
 soundClick2();  
 level++;  
 $("#text").fadeIn(500);  
 $("#text").fadeOut(500);  
 }  
 if(time == 70)  
 {  
 soundClick2();  
 level++;  
 $("#text").fadeIn(500);  
 $("#text").fadeOut(500);  
 }  
 if(time == 80)  
 {  
 soundClick2();  
 level++;  
 $("#text").fadeIn(500);  
 $("#text").fadeOut(500);  
 }  
 if(time == 90)  
 {  
 soundClick2();  
 level++;  
 $("#text").fadeIn(15);  
 $("#text").fadeOut(15);  
 $("#text").fadeIn(15);  
 $("#text").fadeOut(15);  
 $("#text").fadeIn(15);  
 $("#text").fadeOut(15);  
 $("#text").fadeIn(15);  
 $("#text").fadeOut(15);  
 $("#text").fadeIn(15);  
 $("#text").fadeOut(15);  
 $("#text").fadeIn(15);  
 $("#text").fadeOut(15);  
 $("#text").fadeIn(15);  
 $("#text").fadeOut(15);  
 $("#text").fadeIn(15);  
 $("#text").fadeOut(15);  
 $("#text").fadeIn(15);  
 $("#text").fadeOut(15);  
 }  
 time++;  
 document.getElementById("timer");  
}  
  
var backgrounds = //Массив с фонами  
 [  
  
 new Backgrounds("files/images/ground.png", 0),  
 new Backgrounds("files/images/ground2.png", 0),  
 new Backgrounds("files/images/ground3.png", 0),  
 new Backgrounds("files/images/ground4.png", 0),  
 new Backgrounds("files/images/ground5.png", 0),  
 new Backgrounds("files/images/ground6.png", 0),  
 new Backgrounds("files/images/ground7.png", 0),  
 ];  
  
var objects = //Массив игровых объектов  
 [  
 new Basket("files/images/basketImg.gif", 3 \* box, 15 \* box),  
  
 ];  
  
  
function Start()  
{  
 sound3.volume=0.2;  
 sound4.volume=0;  
 sound3.play();  
  
 timer = setInterval(Update, 1000 / 60); //Состояние игры будет обновляться 60 раз в секунду — при такой частоте обновление происходящего будет казаться очень плавным  
  
 SpawnBall();  
}  
  
  
function Update() //Обновление игры  
{  
 /\*backgrounds[0].Update(backgrounds[1]);\*/  
  
 var isDead = false;  
  
 for(var i = 0; i < objects2.length; i++)  
 {  
 objects2[i].Update();  
  
 if(objects2[i].dead)  
 {  
 isDead = true;  
 }  
 }  
  
 if(isDead)  
 {  
 objects2.shift();  
 }  
  
 var hit = false;  
  
 for(var i = 0; i < objects2.length; i++)  
 {  
  
 hit = objects[player].Collide(objects2[i]);  
  
 if(hit)  
 {  
 score ++;  
 hit = false;  
 break;  
 }  
  
 }  
  
 if (live == 0)  
 {  
 sound6.play();  
 /\*alert("Вот и всё, мой дорогой!");\*/  
 StopGame()  
 }  
  
 Draw();  
}  
  
function Draw() //Работа с графикой  
{  
 ctx.clearRect(0, 0, canvas.width, canvas.height); //Очистка холста от предыдущего кадра  
 ctx.drawImage(backgrounds[0].imageBackground, backgrounds[0].xBackground, backgrounds[0].yBackground);  
  
 for (var i = 0; i < objects2.length; i++)  
 {  
 ctx.drawImage(objects2[i].imageBall, objects2[i].xBall, objects2[i].yBall);  
 }  
  
 /\*ctx.drawImage(objects[1].imageBall,objects[1].xBall,objects[1].yBall);\*/  
 ctx.drawImage(objects[player].imageBasket, objects[player].xBasket, objects[player].yBasket);  
 ctx.fillStyle = "white";  
 ctx.font = "50px Arial";  
 ctx.fillText("score", box \* 9.3, box);  
 ctx.fillText(score, box \* 10, box \* 2);  
 ctx.fillText("live", box \* 9.3, box \* 3);  
 ctx.fillText(live, box \* 10, box \* 4);  
 ctx.fillText("time", box \* 9.5 , box \* 5);  
 ctx.fillText(time, box \* 10 , box \* 6);  
 ctx.fillText("level", box \* 9.5, box \* 7);  
 ctx.fillText(level, box \* 10, box \* 8);  
 if (live == 0)  
 {  
 /\* ctx.fillText("Game Over", box \* 2, box \* 2);\*/  
 ctx.drawImage(gameOver, box \* 2, box \* 5);  
 }  
}  
  
  
//Start();  
//init();