Министерство образования и науки Российской Федерации

ФГБОУ ВО «Ульяновский государственный технический университет»

Кафедра «Вычислительная техника»

Лабораторная работа №5

«Игра змейка»

Выполнил студент

группы ИВТАПбд-11

Аронов В. В.

Ульяновск, 2022

**Оглавление**

[**Постановка задачи 3**](#_Toc103984107)

[**Реализация 4**](#_Toc103984108)

[**Интерфейс Игры 8**](#_Toc103984109)

[**Листинг кода 13**](#_Toc103984110)

[**Вывод 15**](#_Toc103984111)

[**Литература 16**](#_Toc103984112)

# **Постановка задачи**

Разаботать на HTML, CSS, JavaScript, +- PHP игру. Тема игры согласуется при сдаче прототипа сайта. Игра должа быть связана с базой данных. Очки полученные в процессе игры должны заносится в базу данных.

# **Реализация**

Лабораторная работа состоит из 3 файлов(game.php, game.css, game.js). В файле game.php расположен блок с самой игрой, вывод результата и кнопка по которой можно обновить рекорд. В game.css прописаны свойства для объектов. Файл game.js состоит из множества функций, основные :

1. Начало игры при загрузке на страницу

function gameLoop() {

    requestAnimationFrame( gameLoop );

…

requestAnimationFrame( gameLoop );

1. Функция обработки нажатия на любую клавишу клавиатуры для изменения направления движения змейки.

document.addEventListener("keydown", function (*e*) {

    if ( *e*.code == "KeyW" ) {

      snake.dy = -config.sizeCell;

        snake.dx = 0;

    } else if ( *e*.code == "KeyA" ) {

        snake.dx = -config.sizeCell;

        snake.dy = 0;

    } else if ( *e*.code == "KeyS" ) {

        snake.dy = config.sizeCell;

        snake.dx = 0;

    } else if ( *e*.code == "KeyD" ) {

        snake.dx = config.sizeCell;

        snake.dy = 0;

    }

});

1. Функция отвечающая за перемещение змейки при пересечении границ canvas-a

function collisionBorder() {

    if (snake.x < 0) {

        snake.x = canvas.width - config.sizeCell;

    } else if ( snake.x >= canvas.width ) {

        snake.x = 0;

    }

    if (snake.y < 0) {

        snake.y = canvas.height - config.sizeCell;

    } else if ( snake.y >= canvas.height ) {

        snake.y = 0;

    }

}

1. Функция отрисовки Яблока(Банана и Вишни)

function drawBerry() {

    var ctx = document.getElementById('game-canvas').getContext('2d');

    var img = **new** *Image*();

    img.onload = function(){

            ctx.drawImage(img,apple.x,apple.y,20,20);

      };

      img.src = '../images/apple.png';

  }

1. Функция рандомного расположения еды, после съедения

function randomPositionBerry() {

    apple.x = getRandomInt( 0, canvas.width / config.sizeCell ) \* config.sizeCell;

    apple.y = getRandomInt( 0, canvas.height / config.sizeCell ) \* config.sizeCell;

}

function randomPositionBanana() {

    banana.x = getRandomInt( 0, canvas.width / config.sizeCell ) \* config.sizeCell;

    banana.y = getRandomInt( 0, canvas.height / config.sizeCell ) \* config.sizeCell;

}

function randomPositionCherry() {

    cherry.x = getRandomInt( 0, canvas.width / config.sizeCell ) \* config.sizeCell;

    cherry.y = getRandomInt( 0, canvas.height / config.sizeCell ) \* config.sizeCell;

}

1. Функция увеличения счета

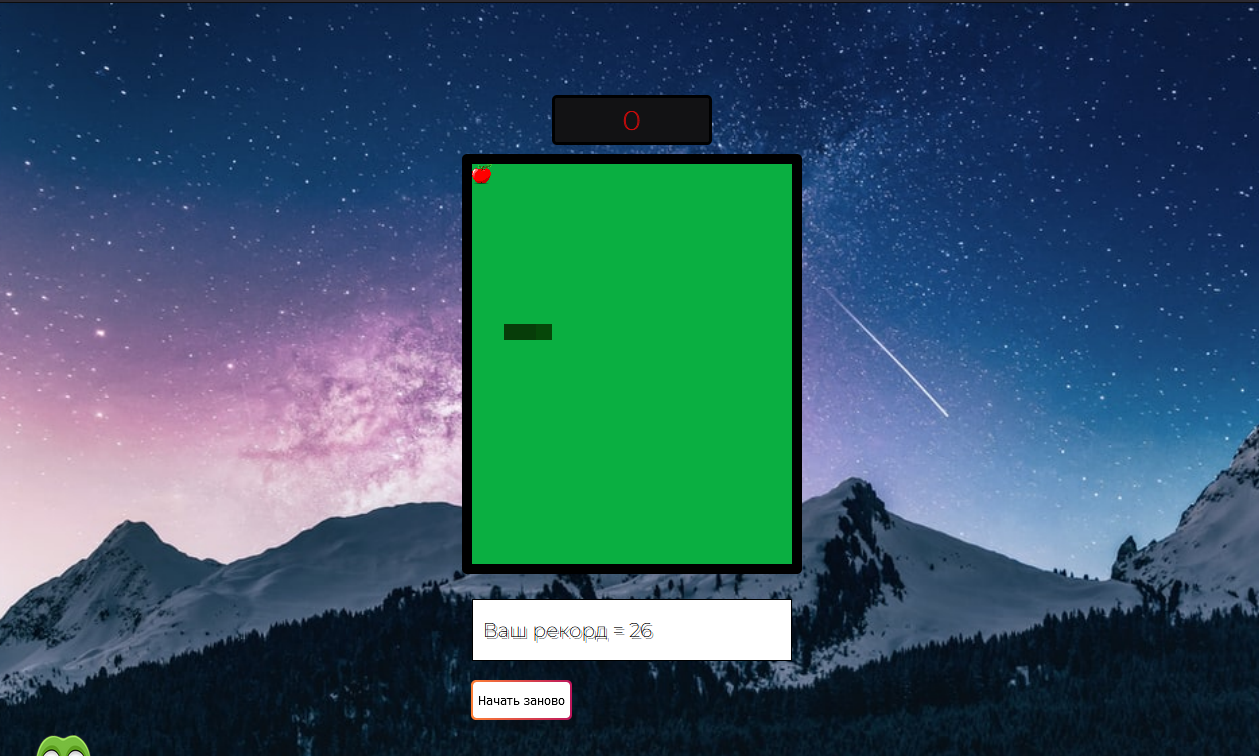
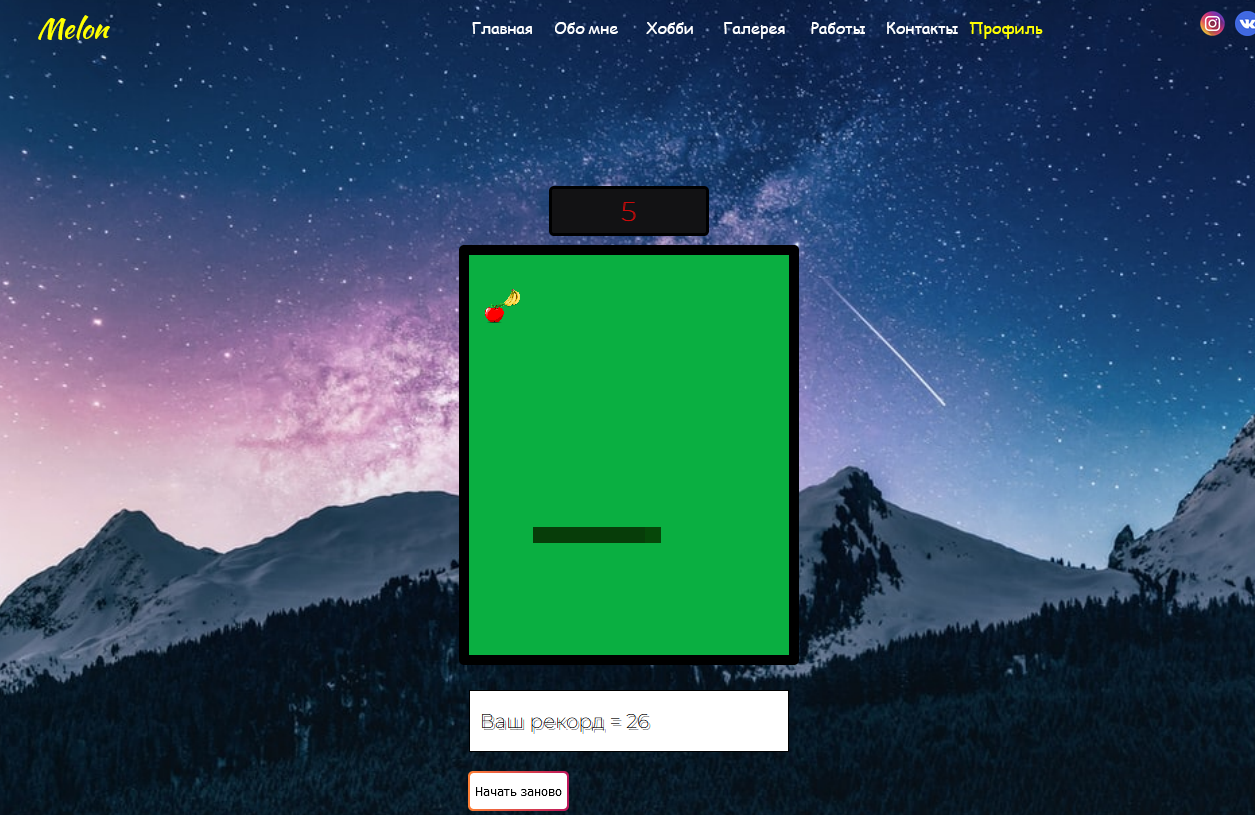
function incScore() {

    score++;

  drawScore();

}

# **Интерфейс Игры**

****Рис 1. – Начало игры, появление змейки и яблока  
  
  
Рис 2. – Появление банана (исчезает после 2.5 секунд)

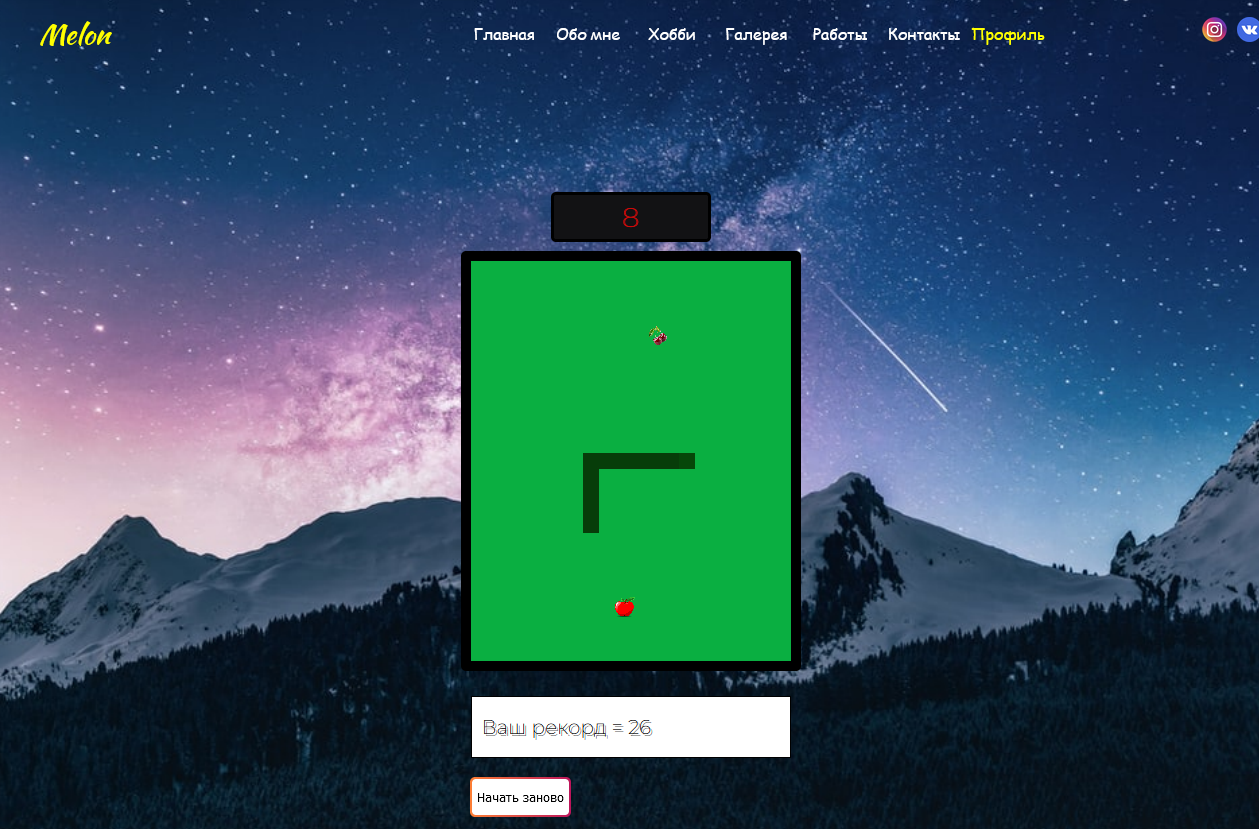


Рис 3. – Появление вишни (исчезает спустя 1.5 секунды)

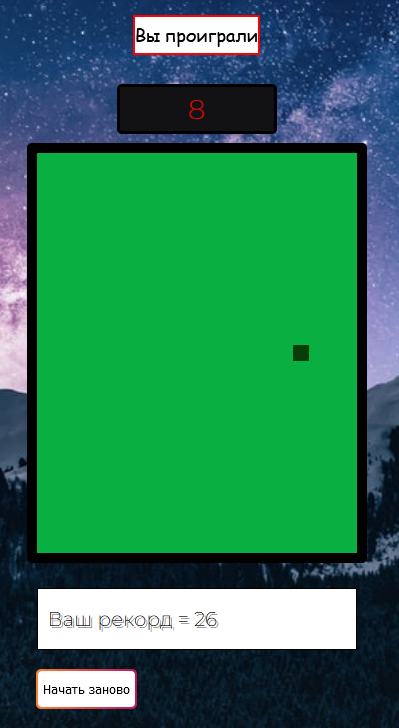
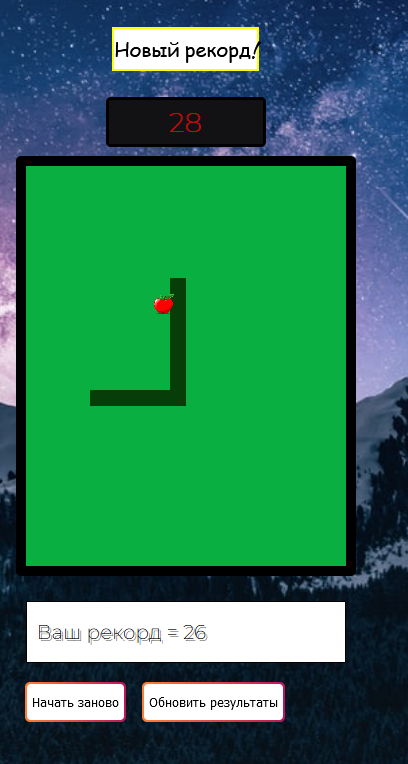
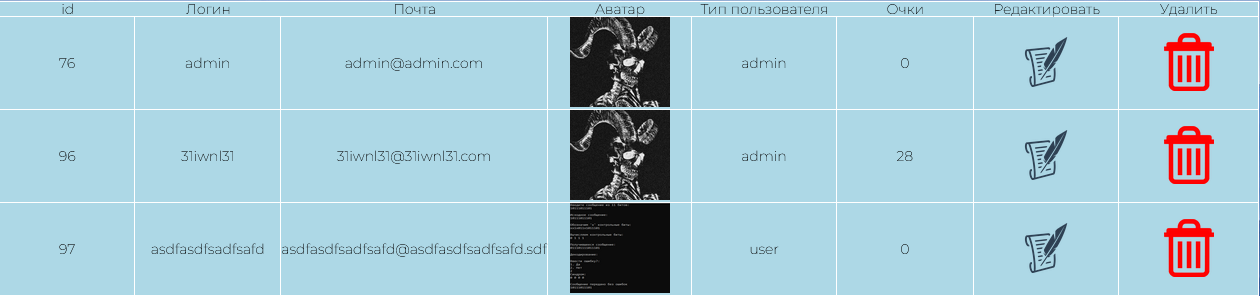


Рис 4. – Вывод результата если пользователь не превзошел предыдущий свой рекорд

  
Рис 5. - Вывод результата если пользователь превзошел предыдущий свой рекорд

  
Рис 6. – Вывод рекорда в таблицу пользователей

# **Листинг кода**

    <div *id*="game">

    <div *class* = "alert" *id* = "death">Вы проиграли</div>

    <div *class* = "alert2" *id* = "new">Новый рекорд!</div>

        <div *class*="game-header">

            <div *class*="game-score">

                <span *class*="score-count">1234</span>

                <p *id*="score"></p>

            </div>

        </div>

        <div *class*="canvas-wrapper">

            <canvas *id*="game-canvas" *width*="320" *height*="400">

                <img *src* = "../images/apple.png" *id* = "lamp" >

            </canvas>

        </div>

game.php

<?php

    session\_start();

    require\_once "openserver.php";

    $id = $\_POST['id'];

    $score = $\_POST['score'];

    mysqli\_query($connect, "UPDATE `users` SET `score` = '$score' WHERE `users`.`id` = '$id'");

    header('Location: ../pages2/game.php');

    ?>

Запрос в бд на редактирование результата

function drawCherry(){

    var ctx = document.getElementById('game-canvas').getContext('2d');

    var img2 = **new** *Image*();

    if(score % 8 == 0 && score != 0 && recoil<15 ){

        cherry.x = 32+ apple.x,

        cherry.y= 64

    img2.onload = function(){

            ctx.drawImage(img2,cherry.x,cherry.y,22,22);

      };

      img2.src = '../images/cherry.jpg';

    }else{

        cherry.x = 500,

        cherry.y = 500

    }

}

function drawBanana(){

    var ctx = document.getElementById('game-canvas').getContext('2d');

    var img3 = **new** *Image*();

    if(score % 5 == 0 && score != 0 && recoil<25 ){

        banana.x = 16+ apple.x,

        banana.y= 32

    img3.onload = function(){

            ctx.drawImage(img3,banana.x,banana.y,22,22);

      };

      img3.src = '../images/banana.png';

    }

    else{

        banana.x = 500,

        banana.y = 500

    }

}

Добавление фруктов (банана и вишни) при определенных очках

# **Вывод**

Благодаря данной лабораторной работе, были приобретены новые навыки работы с бд, а также укреплены старые навыки на языке программирования JavaScript. Получен новый опыт работы с CSS анимацией объектов.

# **Литература**

1. <https://learn.javascript.ru/keyboard-events>
2. <https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API/Tutorial>