Министерство науки и высшего образования Российской Федерации федеральное государственное автономное образовательное учреждение высшего образования

«Санкт-Петербургский политехнический университет Петра Великого»

(ФГАОУ ВО СПбПУ)

Институт компьютерных наук и кибербезопасности

Высшая школа программной инженерии

**КУРСОВАЯ РАБОТА**

По дисциплине «Верстка и прототипирование сайтов»

Разработка игрового приложения «Слова»

(семестр 5)

Студент

группы 3530903/10301 М.В. Волков

Оценка выполненной студентом работы:

Преподаватель,

ст.преподаватель В. Комарова

Санкт-Петербург – 2023

**САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО**

**Институт компьютерных наук и кибербезопасности**

**Высшая школа программной инженерии**

**ЗАДАНИЕ**

на выполнение курсовой работы по дисциплине

«Верстка и прототипирование сайтов»

студенту: Волкову Максиму Вячеславовичу, группа: 5130903/10301

семестр: 5

1. Тема работы: разработка игрового приложения «Слова»
2. Срок сдачи студентом законченной работы: 23.12.2023
3. Исходные данные по работе: материалы по курсу «Верстка и прототипирование сайтов».
4. Содержание работы (перечень подлежащих разработке вопросов):
   1. Введение
   2. Тип игры.
   3. Структура приложения.
   4. Описание работы приложения
   5. Код приложения.
5. Перечень графического материала (с указанием обязательных рисунков): схематизация структуры приложения и порядка выполнения (диаграммы классов UML, диаграммы IDEF0).
6. Дата выдачи задания: «18» ноября 2023 г.

Преподаватель Е.В. Комарова

Студент М.В. Волков

Содержание

[Введение 4](#_Toc156992135)

[1 Архитектура приложения 5](#_Toc156992136)

[2 Игра «Цвета» 7](#_Toc156992137)

[2.1 Описание 7](#_Toc156992138)

[2.2 Логическая структура 7](#_Toc156992139)

[3 Игра «Виселица» 9](#_Toc156992140)

[3.1 Описание 9](#_Toc156992141)

[3.2 Логическая структура 9](#_Toc156992142)

[4 Игра «Найди слова в тексте» 12](#_Toc156992144)

[4.1 Описание 12](#_Toc156992145)

[4.2 Логическая структура 12](#_Toc156992146)

[Заключение 14](#_Toc156992147)

[Список литературы 15](#_Toc156992148)

[Приложение А 16](#_Toc156992149)

Введение

В рамках данной курсовой работы необходимо реализовать игровое приложение, выполняющееся в веб-браузере.

Приложение представляет собой страницу, «размеченную» средствами html и css, и набор функций, разработанных на языке программирования javascript.

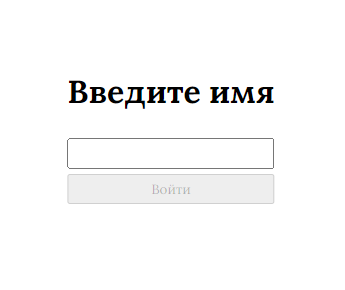
В рамках приложения предполагается разработать несколько игр, так или иначе связанных со словами, их значениями, ошибками в них и т.д. Игры имеют разную механику, поддерживают несколько уровней сложности и систему очков.

1. Архитектура приложения

Приложение состоит из следующих компонентов:

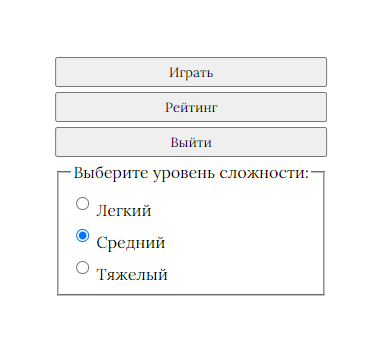
* окно авторизации;
* меню:
* рейтинг игроков;
* игра «Цвета»;
* игра «Виселица»;
* игра «Найди неподходящие слова».

На рисунке 1 изображено окно авторизации.



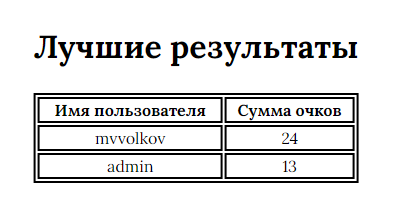
*Рисунок 1 – Окно авторизации*

На рисунке 2 изображено меню.



*Рисунок 2 – Меню*

На рисунке 3 изображен рейтинг игроков.

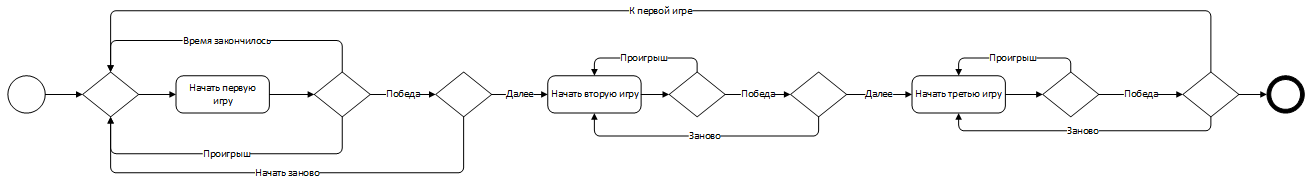


*Рисунок 3 – Рейтинг игроков*

Рейтинг и информация о текущем статусе игр для пользователя хранится в локальном хранилище браузера. При выходе (кнопка «Выйти» в меню) статус пользователя сбрасывается, но рейтинг остается.

Статус пользователя содержит его текущее количество очков, игру, на которой он остановился и текущую страницу. Таким образом, при перезагрузке страницы (или даже перезагрузке браузера, но без очистки хранилища) страница сохранится.

На рисунке 4 представлена схема игры.



*Рисунок 4 – Схема игры*

1. Игра «Цвета»
   1. Описание

В этой игре на странице выводятся блоки с названиями цветов. Каждый блок, помимо названия определенного цвета, имеет окраску. Задача состоит в том, чтобы распределить эти элементы на две категории – название цвета совпадает с цветом элемента или не совпадает.

* 1. Логическая структура

Для распределения элементов по вышеописанным критериям их необходимо перетаскивать в один из контейнеров, расположенных в нижней части страницы, с помощью мыши. Контейнеры подписаны и для удобства окрашены в соответствии с категорией элементов, для которой он предназначен (зеленый контейнер для элементов, в которых название цвета совпадает с окраской, красный – в которых не совпадает).

После перетаскивания элемента в контейнер, он остается там, чтобы после окончания игры была возможность оценить свои результаты, при этом изменить свой выбор после этого уже нельзя.

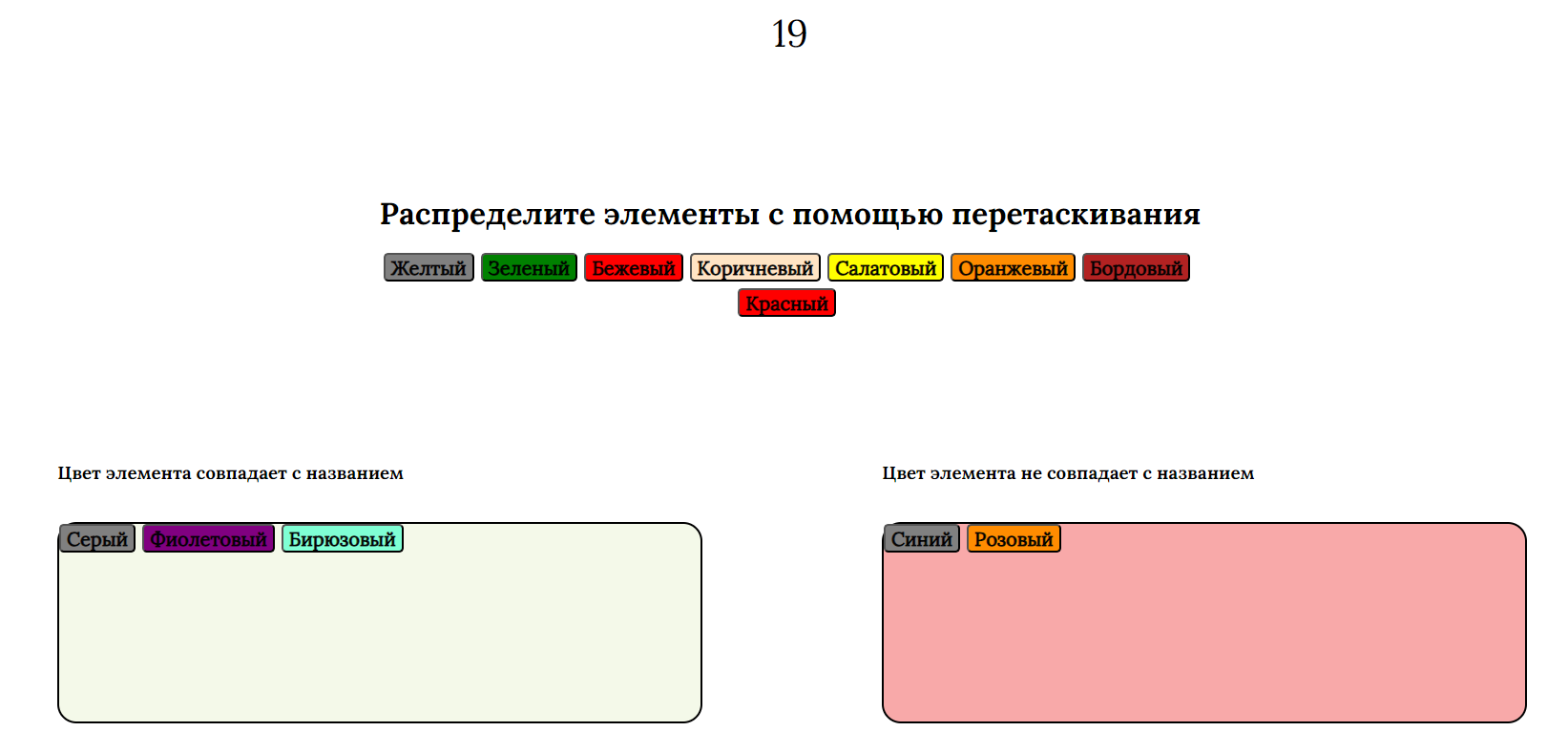
На выполнение задания дается определенное время (оставшееся количество секунд можно увидеть в верхней части страницы), которое зависит от уровня сложности: «легкий» - 59 секунд, «средний» - 40 секунд и «сложный» – 20 секунд. По истечении этого времени попытка считается проваленной (вне зависимости от того, сколько элементов успел распределить пользователь), и будет предложено начать заново или вернуться в меню.

По окончании распределения (если пользователь уложился в заданный временной промежуток) будет произведен подсчет очков – по одному за каждый правильно распределенный элемент. В зависимости от уровня сложности допускается разное количество ошибок: «легкий» - 2 ошибки, «средний» - 1 ошибка и «тяжелый» - ни одной. Если пользователь в процессе игры превышает допустимое для выбранного им уровня сложности количество ошибок, игра завершается преждевременно.

При успешном завершении игры (количество ошибок меньше либо равно допустимому) появляется возможность перейти к следующей игре.

При каждом запуске игры элементы с цветами формируются случайно.

На рисунке 4 представлен ход игры.



*Рисунок 5 – Ход игры «Цвета»*

1. Игра «Виселица»
   1. Описание

В этой игре необходимо угадать слово по буквам, при этом имеется ограниченное число ошибок – пока не завершится рисунок виселицы.

* 1. Логическая структура

В зависимости от уровня сложности выбирается случайное слово из набора (слова для разных уровней сложности отличаются длиной).

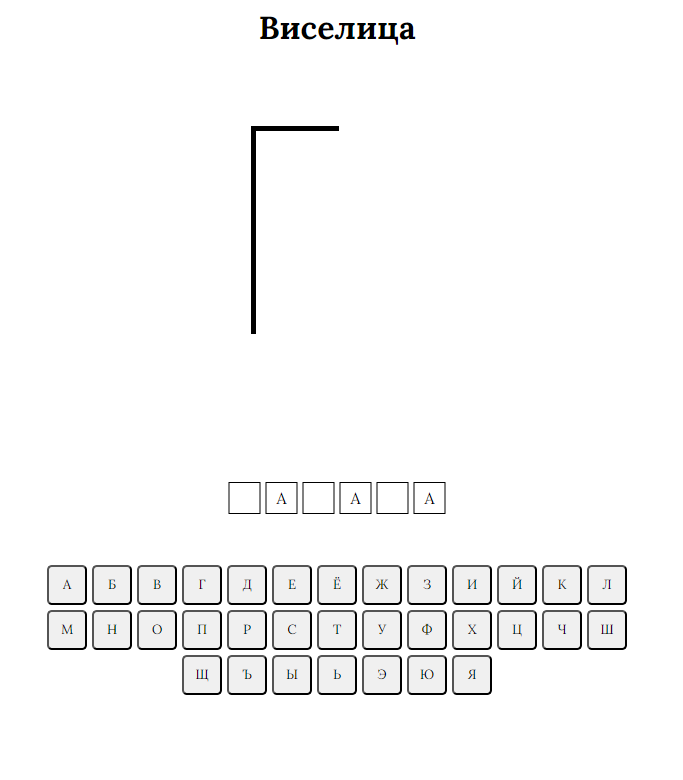
На странице отображаются незаполненные блоки, количество которых соответствует количеству букв в загаданном слове. В нижней части страницы расположены кнопки со всеми буквами алфавита.

Необходимо выбирать буквы, предположительно содержащиеся в слове, с помощью кнопок внизу страницы.

Если выбранная буква присутствует в загаданном слове, то эта буква появляется в блоках на середине экрана в соответствующих местах слова. Если выбранная буква в слове отсутствует, то в верхней половине экрана появляется следующий элемент рисунка с виселицей. Таким образом, при каждой следующей ошибке рисунок становится все более полным и, как только он завершится (9 ошибок), попытка считается проваленной.

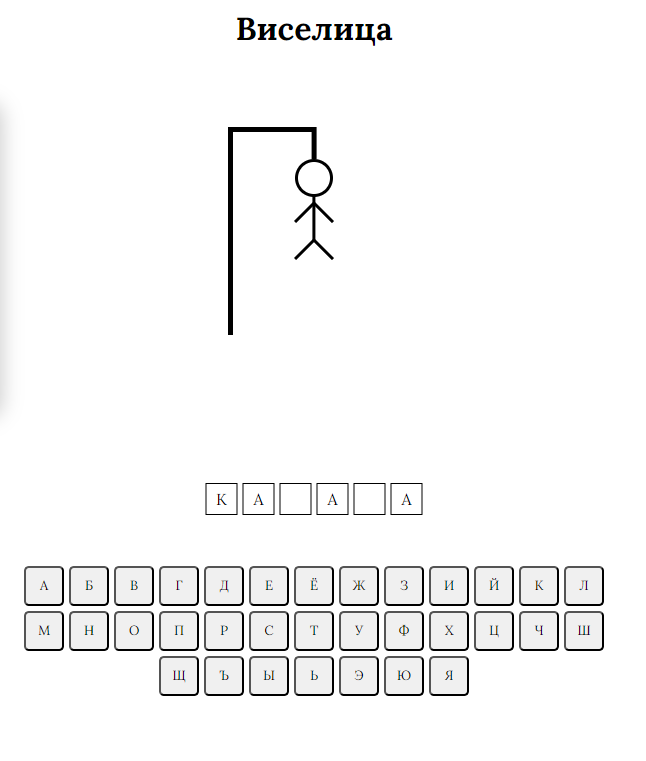
При успешном завершении игры, а именно, если слово угадано, и рисунок виселицы не завершился, появляется возможность перейти к следующей игре.

На рисунке 5 представлен ход игры.



*Рисунок 6 – Ход игры «Виселица»*

На рисунке 6 изображена ситуация, когда картинка виселицы завершена.



*Рисунок 7 – Завершенная картинка «Виселица»*

1. Игра «Найди слова в тексте»
   1. Описание

В этой игре на странице выводится отрывок текста, в котором заменены три слова. Задача состоит в том, чтобы найти эти «неподходящие» слова.

* 1. Логическая структура

При запуске игры выбирается случайный отрывок текста из имеющегося набора.

Для выделения частей текста необходимо зажать левую клавишу мыши, провести до конца предполагаемого выделения и отпустить клавишу. После этого выделенный кусок будет иметь желтый фон.

Редактировать уже созданные области выделения нельзя. Есть возможность очистить все выделения кнопкой «Очистить» внизу. Также нельзя при создании новой области выделения переходить на следующую строку и включать уже имеющиеся области.

При нажатии кнопки «Проверить» правильные области выделения будут иметь зеленый фон, а неправильные - красный. В зависимости от выбранного пользователем уровня сложности отличается минимальное количество слов, которые должны быть отгаданы: «легкий» - 1 слово, «средний» - 2 слова и «тяжелый» - 3 слова (все).

При успешном завершении игры будет предложено попробовать заново, вернуться в меню или к первой игре.

На рисунке 7 представлен результат игры.



*Рисунок 8 – Вариант завершения игры «Найди неподходящие слова»*

Заключение

В ходе данной работы были выбраны или придуманы три игры («Цвета», «Виселица», «Найди неподходящие слова», связанные с взаимодействием со словами, реализовано клиентское веб-приложение, содержащее эти три игры, с возможностью авторизации и рейтингом игроков.

Список литературы

1. Документация Javascript // Mozilla. – 2023. – URL: https://developer.mozilla.org/en-US/docs/Web/JavaScript (дата обращения: 10.12.2023).
2. Документация Javascript // devdocs. – 2023. – URL: https://devdocs.io/javascript (дата обращения: 13.12.2023).
3. Современный учебник Javascript – 2023. – URL: https://learn.javascript.ru (дата обращения: 21.12.2023).

Приложение А

Код приложения

Javascript код для первой игры.

var gameEnded = false;

var words = [

    "Синий", "Красный", "Желтый", "Зеленый", "Оранжевый", "Коричневый",

    "Фиолетовый", "Бежевый", "Серый", "Бирюзовый", "Бордовый", "Розовый", "Салатовый"

]

var colours = [

    "blue", "red", "yellow", "green", "darkorange", "sienna",

    "purple", "bisque", "grey", "aquamarine", "firebrick", "deeppink", "lime"

]

var timeout = null;

var incorrectCount = 0;

function fillWordsButtons() {

    incorrectCount = 0;

    var colouredWordsBox = document.getElementById("coloured-words");

    if (colouredWordsBox.hasChildNodes())

        colouredWordsBox.innerHTML = "";

    var wordsIndices = new Set();

    var unusedWordsIndices = new Set();

    for (var i = 0; i < words.length; i++)

    {

        unusedWordsIndices.add(i);

        wordsIndices.add(i);

    }

    while (unusedWordsIndices.size > 0)

    {

        var wordIndex = getRandomItem(unusedWordsIndices);

        unusedWordsIndices.delete(wordIndex);

        wordsIndices.delete(wordIndex);

        var buttonColourIndex = wordIndex;

        var correct = true;

        if (Math.random() < 0.5)

        {

            buttonColourIndex = getRandomItem(wordsIndices);

            correct = false;

        }

        wordsIndices.add(wordIndex);

        var button = createWordButton(colours[buttonColourIndex], words[wordIndex], correct);

        colouredWordsBox.appendChild(button);

    };

};

function createWordButton(*buttonColour*, *text*, *correct*) {

    var button = document.createElement("button");

    button.classList.add("word-button");

    if (*correct*)

        button.classList.add("correct");

    button.style.background = *buttonColour*;

    button.innerText = *text*;

    button.ondragover = function(*e*) { preventDefaultEventHandling(*e*); }

    button.ondragstart = function(*e*) { *e*.dataTransfer.setData("text", *e*.target.id); }

    button.draggable = "true";

    button.id = uuidv4();

    return button;

}

function drop(*e*, *boxType*) {

    if (gameEnded)

        return;

*e*.preventDefault();

    var button = document.getElementById(*e*.dataTransfer.getData("text"));

    var colouredWordsBox = document.getElementById("coloured-words");

    colouredWordsBox.removeChild(button);

    if (*boxType*)

        document.getElementById("correct-box").appendChild(button);

    else

        document.getElementById("incorrect-box").appendChild(button);

    if (*boxType* != checkDraggedCorrectly(button))

        incorrectCount++;

    if (incorrectCount > getAllowedErrorsCount())

        finish(false);

    if (!colouredWordsBox.hasChildNodes())

        finish(true);

}

function checkDraggedCorrectly(*button*) {

    return *button*.classList.contains("correct");

}

function getAllowedErrorsCount() {

    var difficulty = getState().difficulty;

    if (difficulty === "easy")

        return 2;

    else if (difficulty === "medium")

        return 1;

    else

        return 0;

}

function finish(*success*) {

    clearTimeout(timeout);

    var state = getState()

    var result = `Вы проиграли. Допускается ${getAllowedErrorsCount()} ошибка(и)`;

    var allowNext = false;

    if (*success*)

    {

        allowNext = true;

        victory = true;

        var points = words.length - incorrectCount;

        state.lastTimePointsAdded += points;

        state.points += points;

        setState(state);

        result = `Вы победили! У вас ${points} очков`;

        if (updateStats())

            result += "\nВаш новый рекорд!";

    }

    openModal(result, allowNext);

}

async function launchFirstGameAsync() {

    timerStopRequested = false;

    $("#timer").css("display", "block");

    document.getElementById("correct-box").innerHTML = "";

    document.getElementById("incorrect-box").innerHTML = "";

    var state = getState();

    state.points = 0;

    setState(state);

    victory = false;

    incorrectCount = 0;

    gameEnded = false;

    fillWordsButtons();

    var secondsToFinish = 0;

    var difficulty = state.difficulty;

    if (difficulty === "easy")

        secondsToFinish = 59;

    else if (difficulty === "medium")

        secondsToFinish = 40;

    else

        secondsToFinish = 20;

    timeout = setTimeout(() => {

        $("#timer").css("display", "none");

        gameEnded = true;

        openModal("Время вышло. Попробуйте еще раз", false);

    }, secondsToFinish \* 1000);

    console.log("here");

    await launchTimerAsync(secondsToFinish);

}

Javascript код в для торой игры.

var hangmanWords = {

    "easy": [ "ЦВЕТ", "ДОМ", "ПАСТА", "КРАЙ" ],

    "medium": [ "КАНАВА", "ПИСТОЛЕТ", "ЖИДКОСТЬ", "ПОЛИЦИЯ" ],

    "hard": [ "СИНХРОНИЗАЦИЯ", "ОППОЗИЦИЯ", "СУПЕРПОЗИЦИЯ", "АРАХНОФОБИЯ" ]

}

var hangmanWord = "";

var currentHangmanPart = 1;

function launchSecondGame() {

    currentHangmanPart = 1;

    for (var hangmanPart of document.getElementById("hangman").children)

        hangmanPart.style.display = "none";

    var state = getState();

    hangmanWord = getRandomItem(new Set(hangmanWords[state.difficulty]));

    var wordBox = document.getElementById("word");

    wordBox.innerHTML = "";

    for (var i = 0; i < hangmanWord.length; i++)

    {

        var letterPlaceholder = document.createElement("div");

        wordBox.appendChild(letterPlaceholder);

    }

}

function clickLetter(*button*) {

    if (!hangmanWord.includes(*button*.innerText))

    {

        $(`#hangman${currentHangmanPart}`).css("display", "block");

        currentHangmanPart++;

        if (currentHangmanPart > 9)

            openModal("Вы проиграли. Попробуйте еще раз", false);

        return;

    }

    var letterPlaceholders = [...document.getElementById("word").children];

    for (var i = 0; i < hangmanWord.length; i++)

    {

        if (hangmanWord[i] === *button*.innerText)

            letterPlaceholders[i].innerText = *button*.innerText;

    }

    if (letterPlaceholders.every((*item*) => *item*.innerText != ""))

    {

        var points = 10 - currentHangmanPart;

        var state = getState();

        state.points += points;

        state.lastTimePointsAdded += points;

        setState(state);

        var message = `Вы победили! У вас ${points} очков за уровень и ${state.points} очков всего. `;

        if (updateStats())

            message += "\nВаш новый рекорд!";

        victory = true;

        openModal(message, true);

    }

}

Javascript код для третьей игры.

var texts = [

    `Я всегда твердил, что судьба — игра.<br>

    Что зачем нам солнце, раз есть икра.<br>

    Что готический стиль победит, как школа,<br>

    как способность торчать, избежав осколка.<br>

    Я сижу у окна. За окном осина.<br>

    Я любил многих. Однако — сильно.`,

    `Я считал, что куст — только часть полена.<br>

    Что зачем вся дева, раз есть колено.<br>

    Что, устав от поднятой веком жути,<br>

    русский глаз отдохнет на эстонском шпиле.<br>

    Я сижу у окна. Я помыл посуду.<br>

    Я был несчастлив здесь, и уже не буду.`,

    `Я писал, что в лампочке — ужас пола.<br>

    Что любовь, как акт, лишена существительного.<br>

    Что не знал Эвклид, что, сходя на конус,<br>

    вещь обретает не ноль, но Хронос.<br>

    Я сижу у окна. Приобретаю юность.<br>

    Улыбнусь порою, порой перекрещусь.`,

    `Я сказал, что лист разрушает почку.<br>

    И что семя, упавши в дурную славу,<br>

    не дает побега; что луг с поляной<br>

    есть пример рукоблудья, в Природе данный.<br>

    Я сижу у подоконника, обхватив колени,<br>

    в обществе собственной грузной грусти.`,

    `Моя песня была лишена мотива,<br>

    но зато ее хором не написать. Не диво,<br>

    что в награду мне за такие попытки<br>

    своих ног никто не кладет на плечи.<br>

    Я сижу у окна в темноте; как скорый,<br>

    море гремит за волнистой далью.`

]

var currentTextIndex = 0;

var answers = {

    0: {

        "солнце": "рыба",

        "осколка": "укола",

        "многих": "немногих"

    },

    1: {

        "куст": "лес",

        "жути": "пыли",

        "несчастлив": "счастлив"

    },

    2: {

        "существительного": "глагола",

        "Приобретаю": "Вспоминаю",

        "перекрещусь": "плюнусь"

    },

    3: {

        "славу": "почву",

        "подоконника": "окна",

        "грусти": "тени"

    },

    4: {

        "написать": "спеть",

        "попытки": "речи",

        "далью": "шторой"

    }

}

var selections = []

function launchThirdGame() {

    currentTextIndex = Math.floor(Math.random() \* texts.length);

    setTextToBox();

}

function setTextToBox() {

    $("#text-box").html("<p>" + texts[currentTextIndex] + "</p>");

}

function highlightSelected() {

    var selection = document.getSelection();

    if (selection.isCollapsed)

        return;

    selection = selection.toString();

    var paragraph = document.getElementById("text-box").children[0];

    var currentText = paragraph.innerHTML;

    currentText = currentText.replace(selection, "<span class='user-selection'>" + selection + "</span>");

    currentText = removeNestedSelection(currentText);

    paragraph.innerHTML = currentText;

}

function removeNestedSelection(*text*) {

    var parts = text.split("<span>");

    var finalParts = [parts[0]];

    for (var i = 1; i < parts.length; i++)

    {

        finalParts.push("<span>");

        var innerParts = parts[i].split("</span>");

        finalParts.push(innerParts[0]);

        if (innerParts.length === 1)

           continue;

        for (var j = 1; j < innerParts.length; j++)

        {

            finalParts.push("</span>");

            finalParts.push(innerParts[j]);

        }

    }

    var result = "";

    var highlightStack = 0;

    for (var part of finalParts)

    {

        if (part === "</span>")

        {

            if (highlightStack === 1)

                result += part;

            highlightStack--;

        }

        else if (part === "<span>")

        {

            if (highlightStack === 0)

                result += part;

            highlightStack++;

        }

        else

            result += part;

    }

    return result;

}

function clearSelection() {

    setTextToBox();

}

function checkSelection() {

    var parser = new DOMParser();

    const doc = parser.parseFromString($("#text-box").html(), 'text/html');

    var correctCount = 0;

    var correctHighlights = Object.keys(answers[currentTextIndex]);

    for (var highlight of doc.getElementsByTagName("span"))

    {

        if (correctHighlights.includes(highlight.innerText))

        {

            correctCount++;

            highlight.className = "correct-selection";

        }

        else

            highlight.className = "incorrect-selection";

    }

    $("#text-box").html(doc.documentElement.innerHTML);

    var correctCountNeeded = 0;

    var state = getState();

    if (state.difficulty === "easy")

        correctCountNeeded = 1;

    else if (state.difficulty === "medium")

        correctCountNeeded = 2;

    else

        correctCountNeeded = 3;

    var message = `Вы проиграли. Необходимо отгадать как минимум ${correctCountNeeded}`;

    if (correctCount >= correctCountNeeded)

    {

        state.points += correctCount;

        state.lastTimePointsAdded += correctCount;

        setState(state);

        var message = `Вы победили! У вас ${correctCount} очков за уровень и ${state.points} очков всего. `;

        if (updateStats())

            message += "\nВаш новый рекорд!";

        victory = true;

    }

    openModal(message, victory);

}