**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Радиотехнический»

Кафедра «Информатика и вычислительная техника»

Курс «Парадигмы и конструкции языков программирования»

Отчет по домашнему заданию

Тема «Языки программирования javascript, html, css, python»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы РТ5-31Б: |  | преподаватель каф. ИУ5 |
| Салищев И.Д. |  | Гапанюк Ю.Е. |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2023 г.

**Текст программы с комментариями**

/file clicks.css

/\*данный блок задает шрифт по умолчанию для всего HTML-документа,

Press Start 2P - шрифт с пиксельной обводкой.\*/

html {

    font-family: 'Press Start 2P';

    font-size: 2rem;

}

/\*данный блок выполняет: установку отступов и полей; выравнивание текста по центру или горизонтали;

установку ширины и высоты тела; установку цветов: фон, текст.\*/

body {

    margin: 0;

    padding: 0;

    display: flex;

    justify-content: center;

    width: 100vw;

    height: 100vh;

    background-color: rgb(105, 105, 105);

    color: white;

}

/\*выравнивание + расположение\*/

main {

    display: flex;

    flex-direction: column;

    justify-content: center;

    align-items: center;

}

/\*в данном блоке кода происходит самое интересное!) занимаемся визуалом нашей кнопки!

мы можем поменять размер, цвет, положение, вид курсора при наведении\*/

#button {

    --color: rgb(230, 230, 250);

    --border-color: rgb(230, 230, 250);

    --shadow-color: rgb(50, 55, 61);

    /\*позиционирование\*/

    position: relative;

    /\*уменьшение размера шрифта\*/

    font-size: .75rem;

    /\*размеры кнопки: ширина и высота\*/

    width: 200px;

    height: 100px;

    max-width: 100%;

    max-height: 100%;

    /\*внешние отступы\*/

    margin: 2rem;

    border: none;

    background-size: cover;

    cursor: pointer;

    outline: none;

    color: rgb(75, 75, 85);

    font-family: 'Press Start 2P', monospace;

    font-weight: bold;

    /\*фиксируем статичное положение цветов вокруг кнопки (когда она просто расположена

    на экране)\*/

    background-color: var(--color);

    box-shadow: 0 -3px var(--color),

                3px 0 var(--color),

                0 3px var(--color),

                -3px 0 var(--color),

                0 -6px var(--border-color),

                3px -3px var(--border-color),

                6px 0 var(--border-color),

                0 6px var(--border-color),

                -3px 3px var(--border-color),

                -6px 0 var(--border-color),

                -3px -3px var(--border-color),

                3px 3px var(--border-color),

                0px 9px var(--shadow-color),

                3px 6px var(--shadow-color),

                6px 3px var(--shadow-color),

                0px 9px var(--shadow-color),

                -3px 6px var(--shadow-color),

                -6px 3px var(--shadow-color);

}

#button:active {

    /\*этот блок изменяет стиль кнопки при ее нажатии! (6px - смещает кнопку вниз для эффекта нажатия)\*/

    top: 6px;

    box-shadow: 0 -3px var(--color),

                3px 0 var(--color),

                0 3px var(--color),

                -3px 0 var(--color),

                0 -6px var(--border-color),

                3px -3px var(--border-color),

                6px 0 var(--border-color),

                0 6px var(--border-color),

                -3px 3px var(--border-color),

                -6px 0 var(--border-color),

                -3px -3px var(--border-color),

                3px 3px var(--border-color),

                0px 6px var(--shadow-color),

                3px 3px var(--shadow-color),

                6px 0px var(--shadow-color),

                0px 6px var(--shadow-color),

                -3px 3px var(--shadow-color),

                -6px 0px var(--shadow-color);

}

#display, #counter {

    height: 32px;

}

/file clicks.js

/\*переменная подсчета кликов по кнопке\*/

let clicks = 0;

/\*константа, устанавливающая время игры (в нашем случае 6000 мс = 6 секундам)\*/

const TIMEOUT = 6000;

/\*константы хранящие ссылки на элементы отображения времени, кнопки, счетчика кликов\*/

const display = document.querySelector('#display');

const button = document.querySelector('#button');

const counter = document.querySelector('#counter');

button.onclick = start;

/\*начало\*/

function start() {

    /\*фиксируется время начала\*/

    const startTime = Date.now();

    /\*время игры (в нашем случае 6 сек) отображается на экране\*/

    display.textContent = formatTime(TIMEOUT);

    /\*так называемый счетчик (как работает? при каждом клике на кнопку

    переменная увеличивается и кол-во кликов выводится на экран\*/

    button.onclick = () => counter.textContent = clicks++;

    /\*обновление таймера! каждые 100 мс обновляется отображение оставшегося времени на экране\*/

    const interval = setInterval (() => {

        const delta = Date.now() - startTime;

        display.textContent = formatTime(TIMEOUT - delta);

    }, 100);

    /\*по истечении 6 секунд (6000 мс) игра останавливается;

    вывод сообщения "Игра Завершена"; счетчик времени обнуляется (удаляется)\*/

    const timeout = setTimeout(() => {

        button.onclick = null;

        display.textContent = 'Игра Завершена';

        clearInterval(interval);

        clearTimeout(timeout);

    }, TIMEOUT);

}

/\*функция для преобразования миллисекунд в секунды с 2мя знаками после запятой!

(пр. 3500 мс станут 3.50) \*/

function formatTime(ms) {

    return Number.parseFloat(ms / 1000).toFixed(2);

}

/file clicks.html

<!DOCTYPE html>

<html lang="ru">

    <head>

/\*кодировка документа\*/

        <meta charset="UTF-8">

/\*заголовок страницы, отображаемый на вкладке\*/

        <title>6 секундный кликер</title>

/\*подключение внешнего шрифта\*/

        <link href="https://fonts.googleapis.com/css?family=Press+Start+2P" rel="stylesheet">

/\*подключает наш css файл, который определяет внешний вид эл-тов на стр\*/

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

    </head>

    <body>

        <main>

/\*отображение ост времени\*/

            <div id="display"></div>

/\*кнопка для клика\*/

            <button id="button">клик клик</button>

/\*счетчик кол-ва кликов\*/

            <div id="counter"></div>

        </main>

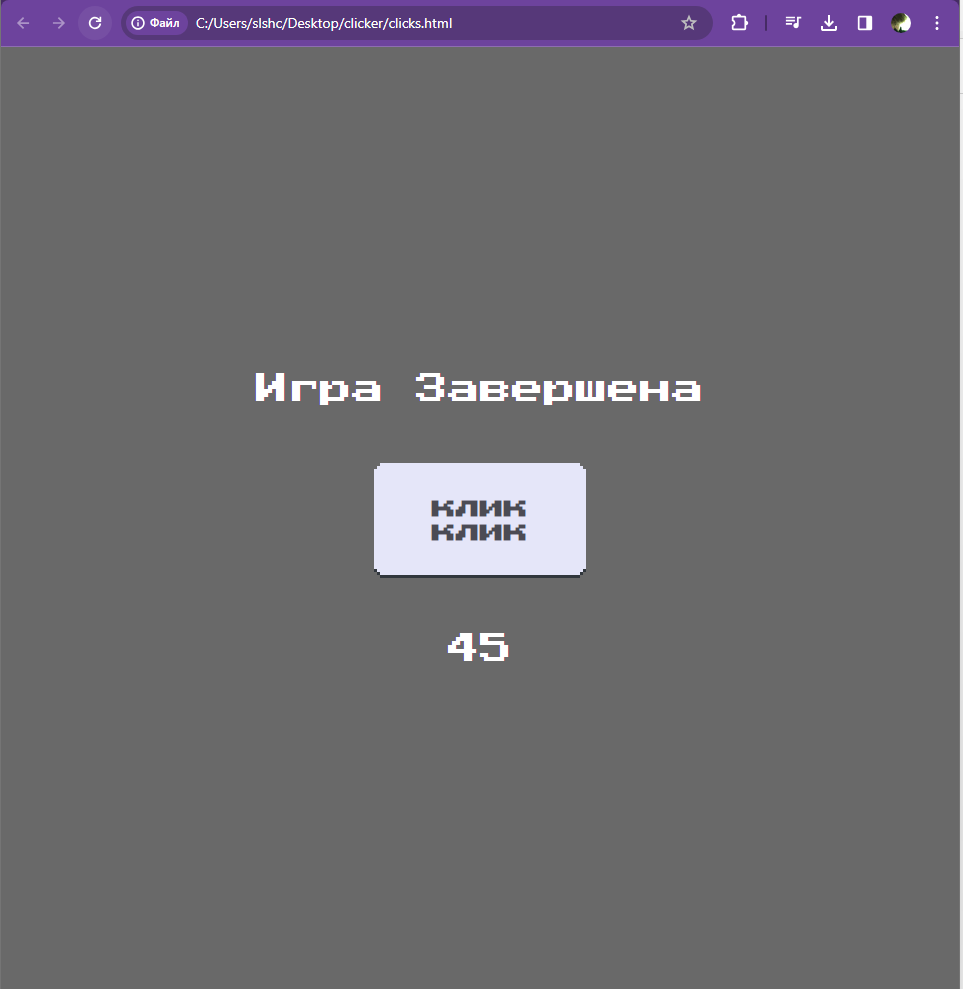
/\*подключает наш js файл, который отвечает за логику игры\*/

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

    </body>

</html>

**Результат**

****

**Дополнительная проверка python**

import keyboard

import mouse

import time

isclick = False

def set\_clicker():

    global isclick

    if isclick:

        isclick = False

        print("кликер откл")

    else:

        isclick = True

        print("кликер вкл")

keyboard.add\_hotkey('Alt + w', set\_clicker)

while True:

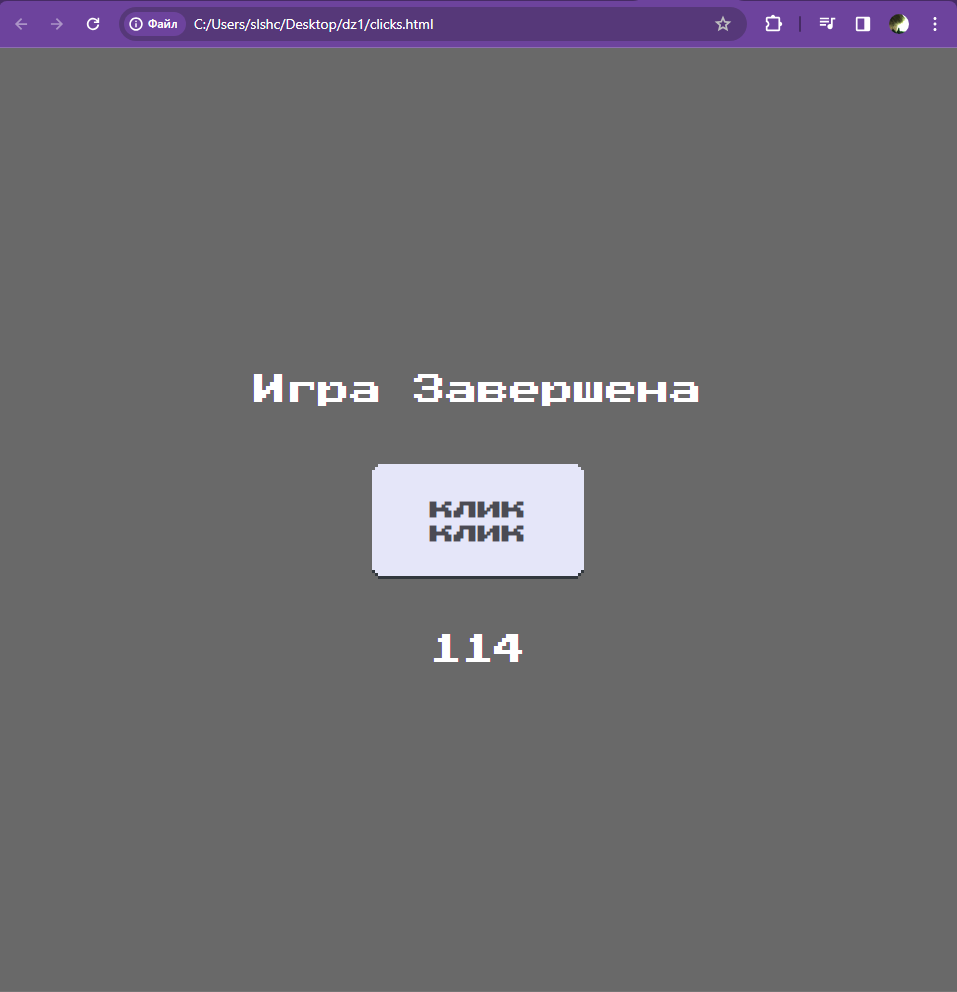
    if isclick:

        mouse.double\_click(button = 'left')

        time.sleep(0.1)

Выше я привел код, который позволяет с задержкой 0.1 с кликать на нашу кнопочку, что можно считать дополнительной проверкой! Для того чтобы запустить проверку на python’e надо установить несколько библиотек: pip install keyboard – позволяет ставить хоткеи и просто взаимодействовать с клавиатурой; pip install mouse – аналогично, как и с клавиатурой; pip install mouse – позволяет взаимодействовать со временем (пр: задержка клика).

**Результат (при помощи python)**

****