Отчет по практической работе №10

Студента группы №81

Лазо Константина Олеговича

**Используя материалы из media.zip (верстку, стили, изображения) реализовать стартовый игровой экран согласно описанию игры в WSR\_2\_MODULE.docx (полной описание игры).**

**Результат:**

Я разбил игру на модули. Реализовал механику переключения сцен: сцена меню переключается на сцену игрового процесса.

Название игры мигает, инструкция анимировано печатается. Кнопка активируется только при наборе никнейма, страница по нажатию на кнопку не перезагружается. Никнейм записывается в переменную.

Класс меню:

import { Scene } from '../abstract-main/scene';

*/\*\**

*\* Класс для сцены меню.*

*\* Обрабатывает блок с классом "screen-start".*

*\*/*

export class Menu extends Scene

{

    constructor(*game*)

    {

**super**(game, 'screen-start');

*// добавляем анимацию*

**this**.animation.flashingText('h1', 'black', 'darkgrey');

**this**.animation.animateText('li', 50);

    }

*// onload*

    init()

    {

**super**.init();

    }

*/\*\**

*\* Функция, работающая постоянно*

*\*/*

    render(*time*)

    {

*// this.game.screen.drawSprite(this.lightBringer);*

*// super.render(time)*

**this**.validateForm()

    }

*/\*\**

*\* Проверяет, что форма не пустая*

*\* Еслти форма не пустая, то активирует кнопку и передаёт nickname*

*\*/*

    validateForm()

    {

        let self = **this**;

        let submit = document.querySelectorAll('.' + **this**.sceneTag + ' > .panel > form > input:last-child');

        let form = document.getElementsByName('nickname');

        switch(form[0].value.length)

        {

            case 0:

                submit[0].style.background = '';

                submit[0].disabled = 'true';

            break

            default:

                submit[0].disabled = '';

                submit[0].style.background = 'green';

                submit[0].onclick = function(*e*)

                {

                    e.preventDefault()

                    self.scene.hidden = 'true';

                    self.finish(Scene.START\_GAME);

                    self.player.nickname = form[0].value;

                }

            break

        }

        submit[0].onmouseout = function() {

**this**.style.background = '';

        }

    }

    getNickname()

    {

    }

}

Класс анимации текста:

*/\*\**

*\* Класс для анимации текста*

*\* Функции:*

*\* anmimateText - анимирует текст, будто его печатают*

*\* flasahingText - мигающий текст*

*\*/*

export class TextAnimation

{

*// аргумент конструктора - это текст ('text'), либо персонаж ('charachter')*

    constructor(*game*)

    {

**this**.game = *game*;

    }

*/\*\**

*\* анимирует текст, будто его печатают*

*\* @param {String} textTag - тэг текста, который надо анимировать*

*\* @param {Int} time - время печати в милисекундах*

*\*/*

    animateText(*textTag*, *time*)

    {

        let tags = document.getElementsByTagName(*textTag*);

        for (let item of tags) {

            let str = item.innerHTML.split("");

            item.innerHTML = '';

            str.forEach(**this**.game.text.tagsOut);

            (function animate()

            {

                let running = setTimeout(animate, *time*);

                str.length > 0 ? item.innerHTML += str.shift() : clearTimeout(running);

            })();

        }

    }

*/\*\**

*\* Анимация мерцающего текста*

*\* @tag Тэг элемента*

*\* @color1 Первый цвет*

*\* @color2 Второй цвет*

*\*/*

    flashingText(*tag*, *color1*, *color2*)

    {

        let self = **this**.game;

        let elm = self.dom.getElementByTag(*tag*);

*//сколько какой цвет по времени*

        let i = 0,

            s = [color2, color1],

            t = [800, 800];

        function animate()

        {

            i ^= 1

            elm.style.color = s[i];

            setTimeout(animate,t[i]);

        }

        animate()

    }

}