Отчет по практической работе №11

Студента группы №81

Лазо Константина Олеговича

**Требуется реализовать движение персонажа с левого края до центра, замирание в центре и передвижение фона при движении героя, передвижение с центра до правого края экрана и в обратном порядке.**

**Результат:**

X игрока изменяется только при движении в начале карты и в конце. При этом через синусоиду я реализовал “движение” игрока по холмам.

Класс сцены игрового процесса, в котором и реализован параллакс-эффект, а также движение по холмам:

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

export class GameProcess extends Scene

{

    constructor(*game*)

    {

**super**(*game*, 'screen-game');

*// интерфейс игрока*

**this**.hud = **this**.game.hud;

**this**.player = **this**.game.player;

*// начальные координаты*

**this**.player.x = 10;

**this**.player.y = **this**.screen.height - 200;

**this**.player.y0 = **this**.screen.height - 200;

        }

**this**.sceneWidth = parseInt(**this**.screen.width \* 6.3);

**this**.background = {

            sceneTag: 'screen-game',

            x: **this**.screen.width / 2

        };

**this**.x = 0;

    }

*// onload*

    init()

    {

**this**.hud.init();

**this**.player.init()

**super**.init();

    }

*/\*\**

*\* Функция, работающая постоянно*

*\*/*

    render(*time*)

    {

**this**.player.regen(*time*)

**this**.hud.render();

**this**.update(*time*);

        let ctx = **this**.screen.context;

        ctx.clearRect(0, 0, **this**.screen.width, **this**.screen.height);

**this**.moving();

**this**.player.init()

*/\*\**

*\* Передвижение по карте, вызов разных состояний (функций) в зависимости от местоположения*

*\* Завершение сцены игры по достижению конца поля*

*\* Проверка, стоит ли игрок*

*\*/*

    moving()

    {

        if(**this**.player.x < **this**.screen.center && **this**.background.x <= **this**.screen.center)

        {

**this**.background.x = **this**.screen.center

            if(**this**.control.right)

            {

**this**.player.standing = true;

**this**.playerMove('right')

            }

            else if(**this**.control.left)

            {

**this**.player.standing = true;

**this**.playerMove('left')

            }

            else if(!**this**.control.right && !**this**.control.left)

            {

**this**.player.standing = true;

            }

        }

*// при приближении игрока к концу поля*

        else if(**this**.background.x >= **this**.sceneWidth)

        {

**this**.background.x = **this**.sceneWidth;

            if(**this**.control.right)

            {

**this**.player.standing = true;

**this**.player.moveRight()

*// завершение игры по достижению конца поля*

                if(**this**.player.x > **this**.screen.width)

                {

                    alert('победа!')

*// self.finish(Scene.START\_GAME);*

                }

            }

            else if(**this**.control.left)

            {

**this**.player.standing = true;

                if(**this**.player.x < **this**.screen.center)

                {

**this**.background.x--;

**this**.backgroundMove('left');

                }

**this**.player.moveLeft()

            }

            else if(!**this**.control.right && !**this**.control.left)

            {

**this**.player.standing = true;

            }

        }

*// параллакс*

        else

        {

            if(**this**.control.right)

            {

**this**.player.standing = false;

**this**.backgroundMove('right')

            }

            else if(**this**.control.left)

            {

**this**.player.standing = false;

**this**.backgroundMove('left')

                if(**this**.player.x >= **this**.screen.center && **this**.background.x <= **this**.screen.center)

                {

**this**.background.x = **this**.screen.center

                    if(**this**.control.left)

                    {

**this**.playerMove('left')

                    }

                }

            }

            else if(!**this**.control.right && !**this**.control.left)

            {

**this**.player.standing = true;

            }

        }

    }

*/\*\**

*\* Движение игрока до центра экрана*

*\* Использует значение функции getX (читать выше),*

*\* которое сравнивает с текщим местоположжением игрока, чтобы определить*

*\* подъёмы и спуски последнего*

*\* @param {string} direction - 'left' или 'right'*

*\*/*

    playerMove(*direction*)

    {

            switch(direction)

            {

                case 'right': **this**.player.moveRight(); break

                case 'left': **this**.player.moveLeft(); break

            }

            let point = **this**.getX(**this**.player)

*// движение по холмам*

            if(point < 150)

            {

*// this.screen.context.setTransform(1, .2, 0, 1, 0, 0);*

                switch(direction)

                {

                    case 'right': **this**.player.y = **this**.player.y + Math.sin(1); break

                    case 'left': **this**.player.y = **this**.player.y - Math.sin(1); break

                }

            }

            else if(point > 150 && point < 500)

            {

**this**.player.y = **this**.player.y;

            }

            else if(point > 500)

            {

                switch(direction)

                {

                    case 'right': **this**.player.y = **this**.player.y - Math.sin(1); break

                    case 'left': **this**.player.y = **this**.player.y + Math.sin(1); break

                }

            }

    }

*/\*\**

*\* В центре экран игрок останавливается, и движение начинает экран*

*\* Параллакс эффект высчитывается по формуле:*

*\* k \* (x - x0)*

*\* k = коэфицент скорости*

*\* x = самая-левая-видимая-координата-фона = (x-координата-фона \* шаг)  / х-координата-центра-экрана)*

*\* x0 = смещение фона вправо изначально*

*\* чем выше итогове значение, тем выше скорость движения*

*\**

*\* @param {string} direction - 'left' или 'right'*

*\*/*

    backgroundMove(*direction*, *k* = 1.7, *step* = **this**.player.step)

    {

        const center = **this**.screen.center;

*// установавливаем для игрока движение вправо*

**this**.player.direction = direction;

**this**.player.init();

*// параллакс (движение фона)*

**this**.scene.style.backgroundPosition = k \* ((**this**.background.x \* step / center) - step) + '%';

*// х-координата фона растёт*

        switch(direction)

        {

            case 'right':

**this**.background.x += **this**.player.step;

**this**.x++;

                break

            case 'left':

**this**.background.x -= **this**.player.step;

**this**.x--;

                break

        }

**this**.player.y = **this**.player.y0 - f(**this**.x);

        function f(*x*)

        {

            let bend = 48 *// изгиб*

*// синусоида, первое число регулирует внутренний угол, второе - выпуклость*

            return Math.sin(x / 60) \* bend;

        }

    }

}