Отчет по практической работе №16

Студента группы №81

Лазо Константина Олеговича

**Используя практическую работу №11-15 дополнить игру 2 возможностями игрока:  
1) Блокировка щитом.  
2) Обычный удар.**

**Описание возможностей (как срабатывают, сколько урона наносят, сколько стоят MP, время на перезарядку и т.д.) использовать из условия в WSR\_2\_MODULE.docx**

**Результат:**

* Основные навыки персонажа (щит и обычная атака) по нажатию на клавиши 1 и 2, соответственно

Абстрактный класс навыка:

*/\*\**

*\* Абстрактный класс навыка (например, удар мечом, щит и т.д.)*

*\* @param {Object} game Класс игры*

*\* @property {HTMLElement} hudElement Элемент из интерфейса с навыком*

*\* @property {Integer} mana Сколько маны потребляет за использование*

*\**

*\*/*

export class Skill

{

    constructor(*game*)

    {

**this**.game = *game*;

**this**.control = **this**.game.control;

**this**.screen = **this**.game.screen;

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

**this**.hud = **this**.game.hud;

**this**.hudElement = '';

*// сколько энергии потребляет*

**this**.mana = 0;

*// таймеры*

**this**.manaDecreaseInterval = 0;

**this**.intervalManaTimer = 0;

*// используется ли в данный момент*

**this**.using = false;

    }

*/\*\**

*\* рендеринг способности*

*\*/*

    render(*time*)

    {

    }

*/\*\**

*\* Ответственная кнопка*

*\*/*

    button()

    {

    }

    decreasePlayerMana(*time*)

    {

        if(**this**.intervalManaTimer == 0)

        {

**this**.intervalManaTimer = *time*

        }

*// console.log(time)*

        if(**this**.intervalManaTimer != 0 && (*time* - **this**.intervalManaTimer) > **this**.manaDecreaseInterval)

        {

            if(**this**.player.mana - **this**.mana < 0)

            {

**this**.player.mana = 0;

            }

            else

            {

**this**.player.mana -= **this**.mana;

            }

**this**.intervalManaTimer = 0;

        }

    }

*/\*\**

*\* Использование способности*

*\*/*

    use(*time*)

    {

*// this.decreasePlayerMana()*

        if(**this**.intervalTimer == 0)

        {

**this**.intervalTimer = *time*

        }

*// if(this.game.control.one && this.intervalTimer != 0 && (time - this.intervalTimer) > this.attackInterval)*

**this**.render(*time*)

*// }*

**this**.intervalTimer = 0;

    }

}

* Смерть героя и врагов

Функции death() и afterDeath() в абстрактном классе персонажа

*/\*\**

*\* Смерть персонажа*

*\*/*

    death()

    {

        if (**this**.health == 0)

        {

**this**.screen.clearRect();

**this**.screen.drawImage(**this**.x, **this**.y, **this**.deathImg, **this**.width, **this**.height);

**this**.afterDeath()

        }

    }

*/\*\**

*\* Что происходит после смерти*

*\*/*

    afterDeath()

    {

    }

Враги пропадают после смерти благодаря функции destruct()

*/\*\**

*\* "Смерть" врага*

*\* Вся его анимация меняется на смерть, позиция фиксируется*

*\*/*

    destruct(*time*)

    {

**this**.rightWalkImg = **this**.deathImg;

**this**.leftWalkImg = **this**.deathImg;

**this**.leftWalkImg = **this**.deathImg;

**this**.attackRightImg = **this**.deathImg;

**this**.attackLeftImg = **this**.deathImg;

**this**.damage = 0;

*// console.log(time)*

        if(**this**.intervalDeathTimer == 0)

        {

**this**.intervalDeathTimer = *time*

        }

        if(**this**.intervalDeathTimer != 0 && (*time* - **this**.intervalDeathTimer) > **this**.deathInterval)

        {

**this**.x = undefined; *// враг пропадает с карты*

        }

    }

* Логика получения удара героем или врагом теперь находится не в классе GameProcess, а в классах персонажа и врага
* Классы врагов теперь создаются не в сцене GameProcess, а в главном классе Game
* и т.д.