**ПРОГРАММИРОВАНИЕ**

**ИНТЕРФЕЙСА**

**3D-СИМУЛЯТОРА**

**УПРАВЛЕНИЕ ДВИЖЕНИЯМИ КАМЕРЫ НА СЦЕНЕ**

Для управления движения камеры относительно установки необходимо, прежде всего, выбрать центральную точку на установке, для чего в скрипте для управления движением камеры нужно перетянуть в добавленном скрипте в окне **Inspector** для камеры центральный объект установки в поле объектной переменной типа **Transform**, например, **targetPos** , что будет определять точку, вокруг которой будет вращаться и двигаться камера.

public class Scroll : MonoBehaviour  
{  
    [SerializeField]  
    float scrollSpeed = 10f;  
    [SerializeField]  
    int sensivity = 3;  
  
    int maxdistance = 20;  
    int mindistance = 1;  
  
    [SerializeField]  
    Transform targetPos;  
  
    void FixedUpdate()  
    {  
        float x = Input.GetAxis("Horizontal"); *// кнопки A D*  
        float y = Input.GetAxis("Vertical"); *// кнопки W S*  
  
        if (x != 0 || y != 0)  
        {  
            Vector3 newpos = transform.position + (transform.TransformDirection(new Vector3(x, 0, 0)) + Vector3.up \* y) / sensivity;  
            if (ControlDistance(Vector3.Distance(newpos, targetPos.position))) transform.position = newpos;  
        }  
  
        if (Input.GetAxis("Mouse ScrollWheel") != 0)  
        {  
            Vector3 newpos = transform.position + transform.TransformDirection(Vector3.forward \* Input.GetAxis("Mouse ScrollWheel") \* scrollSpeed);  
            if (ControlDistance(Vector3.Distance(newpos, targetPos.position)))   
            transform.position = newpos;  
        }  
         
        if (Input.GetMouseButton(1))  
        {  
            transform.RotateAround(targetPos.position, Vector3.up, Input.GetAxis("Mouse X")\*sensivity);  
           *// transform.Rotate(Vector3.left\*Input.GetAxis("Mouse Y")\*sensivity);*  
        }  
    }  
  
    bool ControlDistance (float distance)  
    {  
        if (distance > mindistance && distance < maxdistance) return true;  
        return false;  
    }

**СКРИПТ ДЛЯ ПЕРЕМЕЩЕНИЯ КАМЕРЫ**

**public class Scroll : MonoBehaviour**

**{**

**[SerializeField]**

**Transform targetPos;**

**int sensivity = 3;**

**int maxdistance = 20;**

**int mindistance = 1;**

// ФУНКЦИЯ ОГРАНИЧЕНИЯ ПРЕДЕЛОВ ДВИЖЕНИЯ КАМЕРЫ

**bool ControlDistance (float distance)**

**{**

**if (distance > mindistance && distance < maxdistance) return true;**

**return false;**

**}**

// ВРАЩЕНИЕ ВОКРУГ ЦЕНТРАЛЬНОЙ ТОЧКИ УСТАНОВКИ С ЗАЖАТОЙ ПРАВОЙ КЛАВИШЕЙ МЫШИ

**void Update()**

**{**

**if (Input.GetMouseButton(1))**

**{**

**transform.RotateAround(targetPos.position, Vector3.up, Input.GetAxis("Mouse X")\*sensivity);**

**}**

// ДВИЖЕНИЯ КАМЕРЫ В СТОРОНЫ КЛАВИШАМИ

**float x = Input.GetAxis("Horizontal");**

**float y = Input.GetAxis("Vertical");**

**if (x != 0 || y != 0)**

**{**

**Vector3 newpos = transform.position + (transform.TransformDirection(new Vector3(x, 0, 0)) + Vector3.up \* y) / sensivity;**

**if (ControlDistance(Vector3.Distance(newpos, targetPos.position))) transform.position = newpos;**

**}**

// ПРИБЛИЖЕНИЕ И УДАЛЕНИЕ КАМЕРЫ ОТ УСТАНОВКИ ПРОКРУТКОЙ КОЛЕСА МЫШИ

**if (Input.GetAxis("Mouse ScrollWheel") != 0)**

**{**

**Vector3 newpos = transform.position + transform.TransformDirection(Vector3.forward \* Input.GetAxis("Mouse ScrollWheel") \* scrollSpeed);**

**if (ControlDistance(Vector3.Distance(newpos, targetPos.position))) transform.position = newpos;**

**}**

**}**

**}**