**РАЗРАБОТКА ИНТЕРФЕЙСА ДЛЯ СИМУЛЯТОРА УСТАНОВКИ**

**АЛГОРИТМ ПЕРЕДВИЖЕНИЯ КАМЕРЫ В ЗАДАННУЮ ТОЧКУ ДЛЯ УДОБНОГО ПРОСМОТРА ОБЪЕКТА:**

Для получения положения камеры в нужном ракурсе для просмотра соответсвующего элемента установки можно поступать следующим образом:

1. Запомнить значения позиции **Position** и поворота **Rotation** камеры в исходном положении в окне редактора среды **Inspector**.
2. Рассчитать (рационально это сделать вручную на отдельном чертеже-схеме) желаемую позицию и поворот камеры для удобного ракурса просмотра соответсвующего элемента установки – сдвиг рассчитывается как разность значений позиций XYZ между установкой и камерой, а поворот камеры на нужный угол определяется в конструкции кватерниона.

Например, для просмотра элемента на обратной стороне установки требуемый угол поворота будет равен:

**needRotaton = Quaternion.AngleAxis(-180, new Vector3(0,1,0));**

а для просмотра сверху:

**needRotaton = Quaternion.AngleAxis(0, new Vector3(0,1,0));**

1. В скрипте для перемещения камеры в нужные ракурсы задать для каждой кнопки отдельные функции.

Например, для просмотра обратной стороны установки задать функцию **Move()**:

**public void Move()**

**move = true;**

**startPosition = transform.position;**

**startRotation = transform.rotation;**

**needPosition = new Vector3(277.0f, 251, 6);**

**needRotaton = Quaternion.AngleAxis(-180, new Vector3(0,1,0));}**

Здесь булевая переменная **move**используется для запуска движения, а стартовый позиция и угол поворота задаются каждый раз для исходного положения камеры перед ее перемещением.

1. Обеспечить движение камеры от исходного положения к точке с нужным ракурсом просмотра элемента установки с использованием метода **Update()**

**СКРИПТ ПЕРЕДВИЖЕНИЯ КАМЕРЫ В ЗАДАННУЮ ТОЧКУ ДЛЯ УДОБНОГО ПРОСМОТРА ОБЪЕКТА:**

**public class replacer1 : MonoBehaviour  
{      bool move = false;  
        float speed = 0.01f;  
        float offset = 0;  
        Vector3 startPosition;  
        Vector3 needPosition;  
        Quaternion startRotation;  
        Quaternion needRotaton;  
       public void Move1()** //функция для просмотра обратной стороны установки **{  move = true;  
            startPosition = transform.position;  
            startRotation = transform.rotation;  
            needPosition = new Vector3(277.0f, 251, 6);  
            needRotaton = Quaternion.AngleAxis(-180, new Vector3(0,1,0));        }  
        public void Move2()** //функция для просмотра установки справа **{  move = true;  
            startPosition = transform.position;  
            startRotation = transform.rotation;  
            needPosition = new Vector3(284, 253, 0f);  
            needRotaton = Quaternion.AngleAxis(-90, new Vector3(0,1,0));        }  
        public void Move3()** //функция для просмотра установки сверху **{   move = true;  
            startPosition = transform.position;  
            startRotation = transform.rotation;  
            needPosition = new Vector3(277, 255, 0f);  
            needRotaton = Quaternion.AngleAxis(90, new Vector3(1,0,0));        }  
        public void Move0()** //функция для возврата к первоначальному виду установки **{   move = true;  
            startPosition = transform.position;  
            startRotation = transform.rotation;  
            needPosition = new Vector3(276.12f, 252.9f, -6.93f);  
            needRotaton = Quaternion.AngleAxis(0, new Vector3(0,1,0));        }**

**void Update()  
        {   if(move)  
            { offset+=speed;  
            transform.position = Vector3.Lerp(startPosition, needPosition, offset);  
            transform.rotation = Quaternion.Slerp(startRotation, needRotaton, offset);              
             if (offset >= 1)  
                { move = false;  
                offset = 0;      }  
            }  
        } }**

В последней процедуре скрипта обеспечивается движение камеры к точке с нужным ракурсом просмотра элемента установки, при этом возрастающая скорость перемещения обеспечивается переменной **offset** с ограничением ее значения до 1 для прекращения движения – **move = false.**