

# **Руководство пользователя**

Инструментарий интерактивных элементов управления для внедрения на сцену 3D проекта Unity, созданный с целью ускорения и упрощения создания проектов, предполагающих взаимодействие с пользователем, как в версии для персонального компьютера, так и для устройств виртуальной реальности, а также библиотека моделей и текстур.

## Содержание

1. Скрипт для размещения и настройки .....	3
2. Кнопка .....	6
2.1. Модели .....	6
2.2. Скрипт .....	6
3. Рычаг .....	9
3.1. Модели .....	9
3.2. Скрипт .....	9
4. Вентиль .....	12
4.1. Модели .....	12
4.2. Скрипты .....	12
5. Ползунок .....	15
5.1. Модели .....	15
5.2. Скрипты .....	15
6. Переключатель .....	18
6.1. Модели .....	18
6.2. Скрипты .....	18
7. Визуальное подсвечивание .....	21

## 1. Скрипт для размещения и настройки

Данный скрипт на C# представляет компонент для размещения и настройки элементов на сцене в Unity 3D. Он позволяет выбрать и разместить объекты на сцене, изменять их масштаб и вращение, а также сбрасывать настройки.

### **Добавление скрипта:**

- 1) Создайте новый пустой объект или выберите существующий объект в сцене Unity.
- 2) Прикрепите скрипт "PlacementScript" к выбранному объекту.
- 3) Установите необходимые значения для публичных переменных и настройте другие параметры по вашему выбору.

### **Настройка публичных переменных:**

- 1) `placeableObjectPrefabs`: Массив префабов игровых объектов, которые можно размещать.
- 2) `PlacementDone`: Флаг, указывающий, завершено ли размещение объекта.

В скрипте используется техника лучевой трассировки (`raycast`) для размещения объектов в сцене. Поэтому, перед тем как добавлять префабы в массив `placeableObjectPrefabs`, необходимо установить слой "Ignore Raycast" для всех элементов внутри каждого префаба.

- 3) `ModePrefabCount`: Количество префабов в каждом режиме (режим предназначен для выбора различных групп префабов).
- 4) `ModeSelected`: Флаг, указывающий, выбран ли режим размещения.

### **Взаимодействие с объектами:**

- 1) При выборе режима размещения (нажатие клавиш `1 - (ModePrefabCount + 1)`), будет активирован соответствующий режим и установлен `modeParameter`.
- 2) Для размещения объектов используйте клавишу, соответствующую порядковому номеру объекта в текущем режиме (например, для размещения первого объекта в режиме 3, используйте сначала клавишу «3», а после – клавишу «1»).
- 3) Если объект уже в процессе размещения и использована клавиша для того же самого префаба, объект будет удален.
- 4) После размещения объекта, щелчок левой кнопкой мыши завершит процесс размещения (`PlacementDone` будет установлен в `true`).

5) Чтобы удалить размещаемый объект и сбросить режим, используйте клавишу Backspace.

#### **Управление объектами:**

- 1) Чтобы переместить текущий объект курсора мыши, он будет следовать за положением курсора на плоскости, коллизия с которой определяется лучом от камеры.
- 2) Для изменения масштаба объекта используйте клавиши "+" и "-" на клавиатуре.
- 3) Для вращения объекта с помощью колесика мыши используйте прокрутку колесика мыши.
- 4) Чтобы сбросить настройки объекта (масштаб, вращение) к значениям по умолчанию, нажмите правую кнопку мыши.

В ассете доступны 5 видов элементов, каждый из которых имеет 3 модели. Таким образом,  $\text{ModePrefabCount} = 3$ , и используются клавиши от 1 до 5 для выбора режима и клавиши от 1 до 3 для выбора префаба.

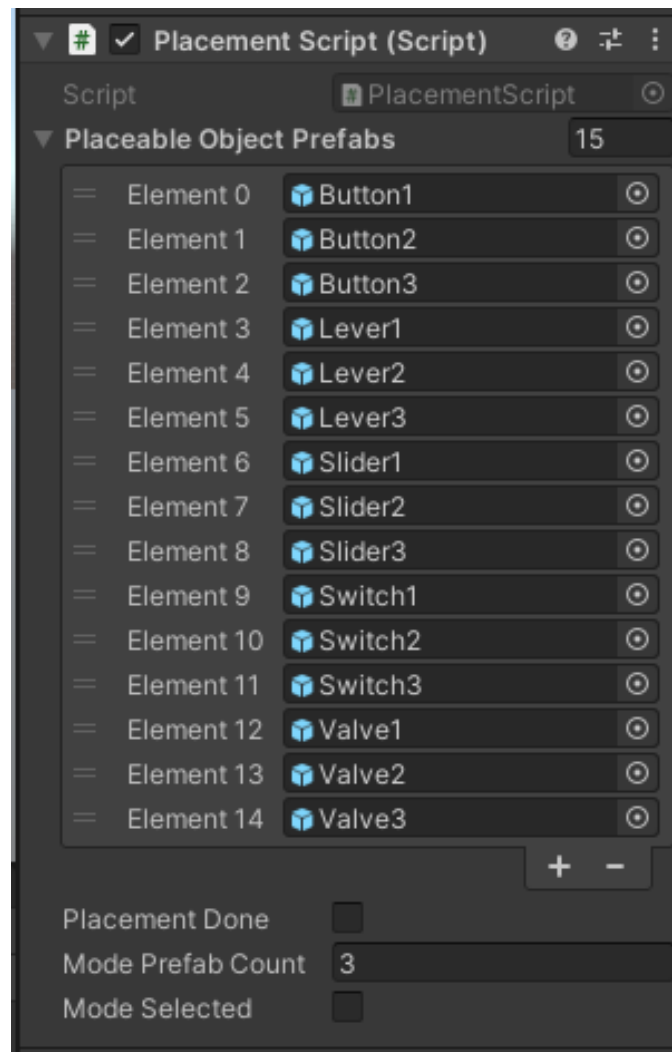


Рис.1.1. Скрипт PlacementScript.cs в редакторе Unity

## 2. Кнопка

### 2.1. Модели

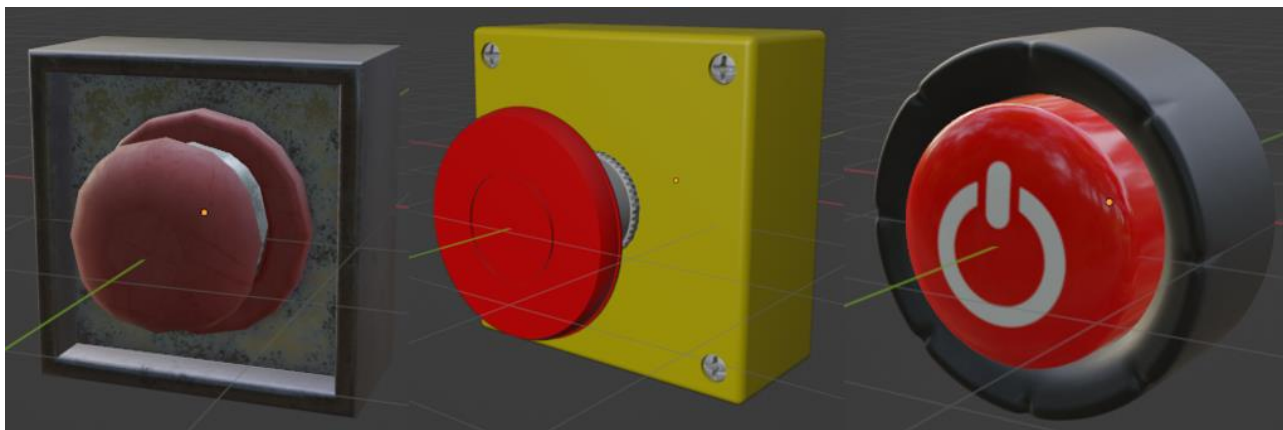


Рис.2.1. 3D-модели для интерактивного элемента «Кнопка»

### 2.2. Скрипт

#### **Подготовка модели:**

- 1) Необходимо наличие двух моделей: Одной моделью является основание элемента, а второй моделью — ручка элемента, с которой пользователь будет взаимодействовать.
- 2) Необходима настройка rigid body для обеих моделей: Необходимо установить режим rigid body (жесткое тело) для обеих моделей. Это позволяет моделям взаимодействовать механически и сохранять свою форму и положение в пространстве.
- 3) Необходима настройка позиции точки вращения – pivot point – для обеих моделей:
  - a. У основания элемента: Должна быть минимальной по оси y и находиться в середине по осям x и z.
  - b. У ручки элемента: Должна быть в центре объекта.

#### **Добавление скрипта:**

- 1) Создайте новый пустой объект или выберите существующий объект в сцене Unity.
- 2) Прикрепите скрипт "Button" к выбранному объекту.
- 3) Установите необходимые значения для публичных переменных и настройте другие параметры по вашему выбору.

#### **Настройка публичных переменных:**

- 1) myGameObject: Ссылка на игровой объект, который будет изменяться при нажатии кнопки.
- 2) elementHandle: Ссылка на игровой объект, представляющий верхнюю часть кнопки.
- 3) buttonLowerLimit: Нижняя позиция кнопки (по оси Y), к которой она будет опускаться при нажатии.
- 4) Force: Сила, с которой кнопка будет двигаться вниз при удержании клавиши.
- 5) Threshold: Пороговое значение, определяющее максимальное расстояние, на которое кнопка может быть нажата.

### **События:**

- 1) onPressed: Событие, которое будет вызываться при нажатии кнопки.
- 2) onReleased: Событие, которое будет вызываться при отпускании кнопки.

### **Режимы работы:**

- 1) changeModeButtonName: Название клавиши, которая будет использоваться для переключения режима работы (обычный режим или VR-режим).
- 2) isVRMode: Флаг, определяющий текущий режим работы кнопки.

### **Взаимодействие с кнопкой посредством клавиатуры:**

Для нажатия кнопки используется клавиша, указанная в pressButtonName.

### **Взаимодействие с кнопкой в VR-режиме:**

- 1) При включении VR-режима кнопка будет реагировать на взаимодействие с помощью VR-контроллера.
- 2) При захвате кнопки VR-контроллером она будет перемещаться в зависимости от положения контроллера.
- 3) При отпускании кнопки она вернется в исходную позицию.

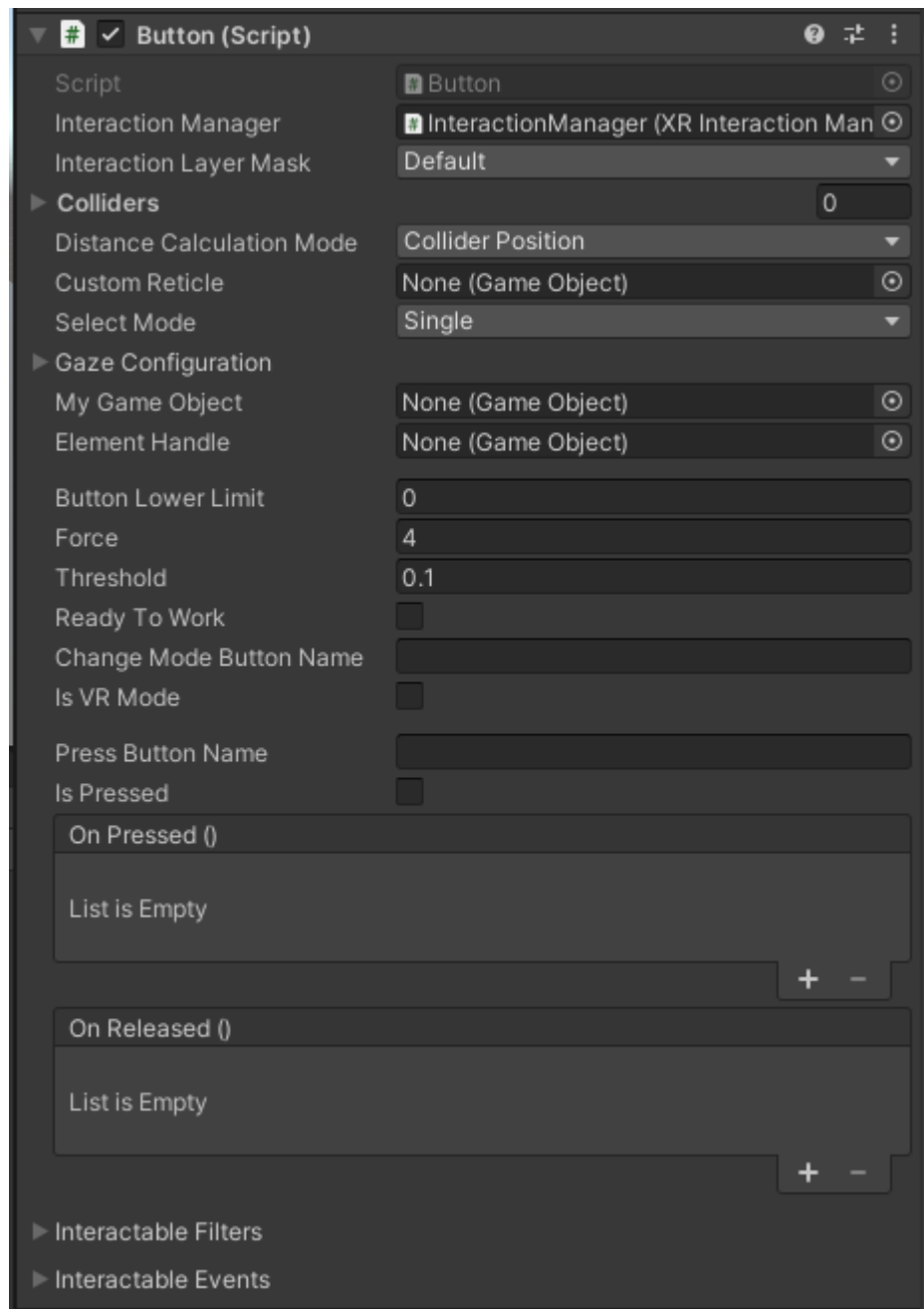


Рис.2.2. Скрипт Button.cs в редакторе Unity



### 3. Рычаг

#### 3.1. Модели

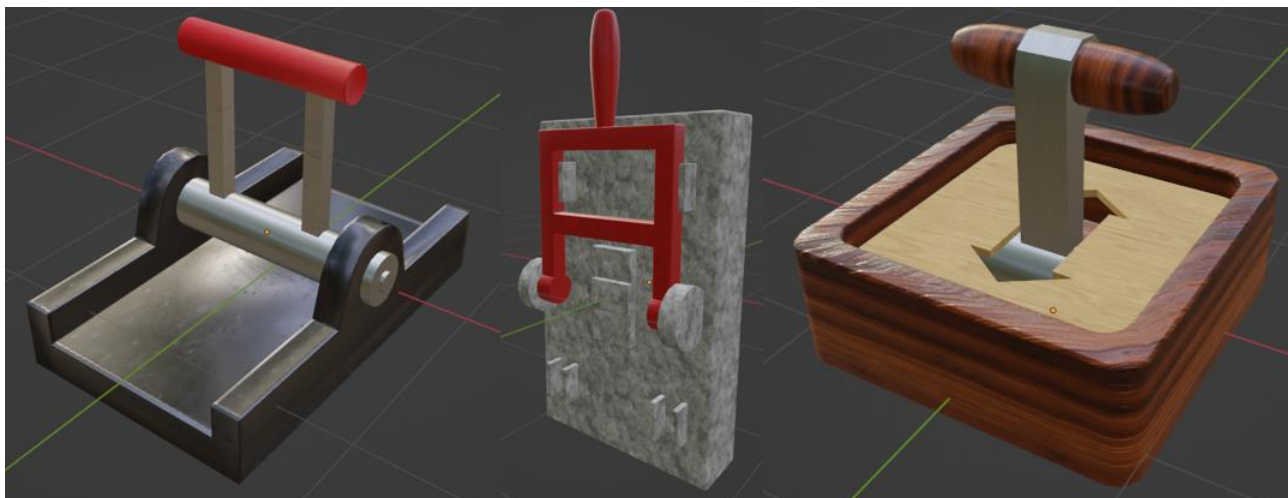


Рис.3.1. 3D-модели для интерактивного элемента «Рычаг»

#### 3.2. Скрипт

##### Подготовка модели:

- 1) Необходимо наличие двух моделей: Одной моделью является основание элемента, а второй моделью — ручка элемента, с которой пользователь будет взаимодействовать.
- 2) Необходима настройка rigid body для обеих моделей: Необходимо установить режим rigid body (жесткое тело) для обеих моделей.
- 3) Необходима настройка позиции точки вращения – pivot point – для обеих моделей:
  - a. У основания элемента: Должна быть минимальной по оси y и находиться в середине по осям x и z.
  - b. У ручки элемента: Должна быть в центре вращения ручки.
- 4) Так как объект требует вращения в процессе взаимодействия, необходимо настроить hinge joint: Для обеспечения возможности вращения объекта вокруг оси, в документации необходимо описать настройку hinge joint.

##### Добавление скрипта:

- 1) Создайте новый пустой объект или выберите существующий объект в сцене Unity.
- 2) Прикрепите скрипт "Lever" к выбранному объекту.
- 3) Установите необходимые значения для публичных переменных и настройте другие параметры по вашему выбору.

### **Настройка публичных переменных:**

- 1) myGameObject: Ссылка на игровой объект, который будет изменяться при использовании рычага.
- 2) elementHandle: Ссылка на игровой объект, представляющий рычаг.
- 3) AngleMin: Минимальный угол поворота рычага.
- 4) AngleMax: Максимальный угол поворота рычага.
- 5) Threshold: Пороговое значение, определяющее максимальное расстояние от минимального или максимального угла поворота, при котором считается, что рычаг нажат.
- 6) rotationSpeed: Скорость поворота рычага при удержании клавиш.

### **События:**

- 1) onPressedMin: Событие, которое будет вызываться при нажатии рычага в минимальную позицию.
- 2) onPressedMax: Событие, которое будет вызываться при нажатии рычага в максимальную позицию.
- 3) onReleased: Событие, которое будет вызываться при отпускании рычага.
- 4) onValueChange: Событие, которое будет вызываться при изменении значения рычага. Возвращает текущий процент поворота рычага.

### **Режимы работы:**

- 1) changeModeButtonName: Название клавиши, которая будет использоваться для переключения режима работы (обычный режим или VR-режим).
- 2) isVRMode: Флаг, определяющий текущий режим работы рычага.

### **Взаимодействие с рычагом посредством клавиатуры:**

Для нажатия рычага используются клавиши, указанные в pressMinButtonName и pressMaxButtonName.

### **Взаимодействие с рычагом в VR-режиме:**

Рычаг может быть нажат путем захвата и перемещения контроллера в VR-режиме.

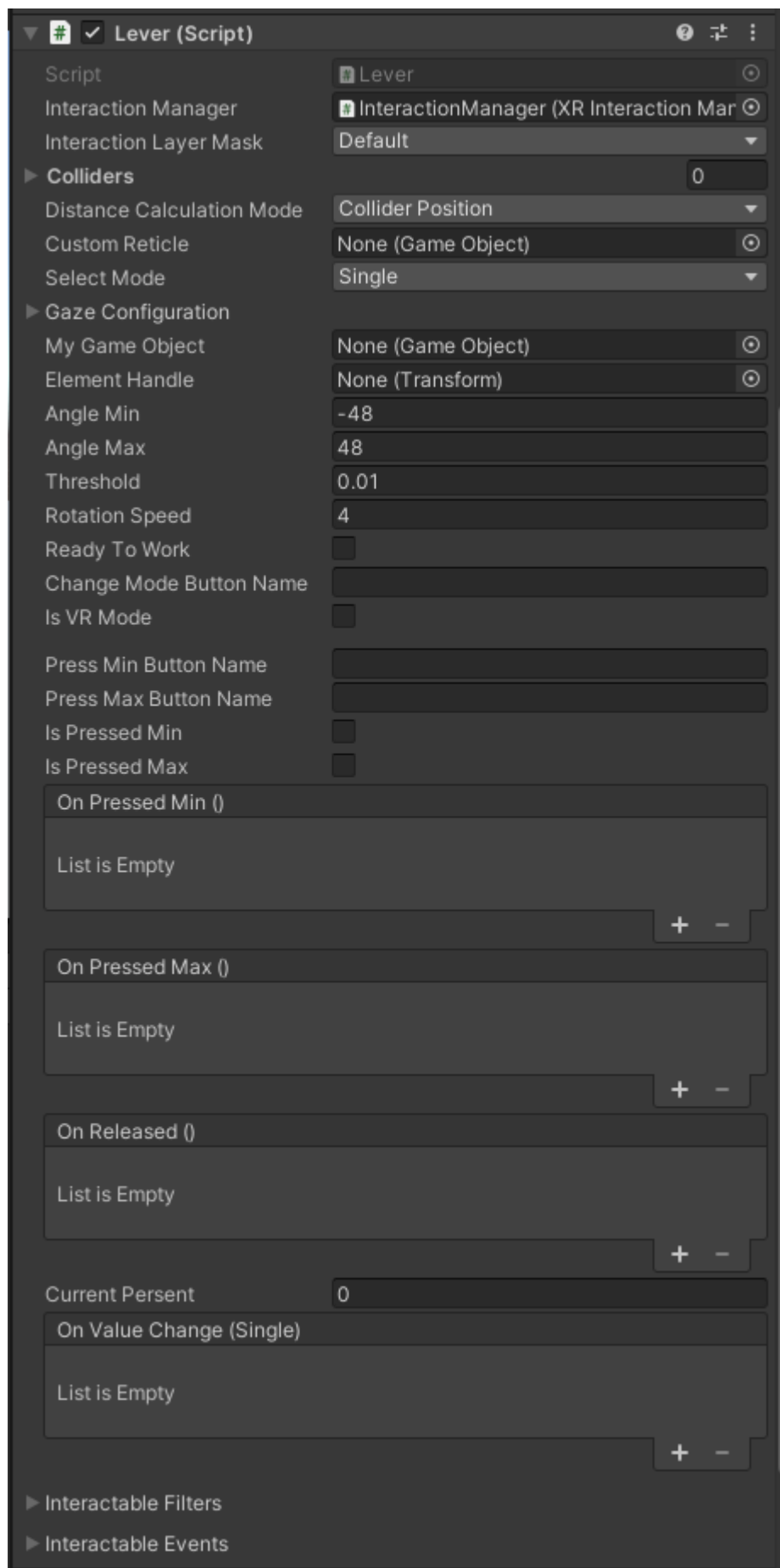


Рис.3.2. Скрипт Lever.cs в редакторе Unity

## 4. Вентиль

### 4.1. Модели

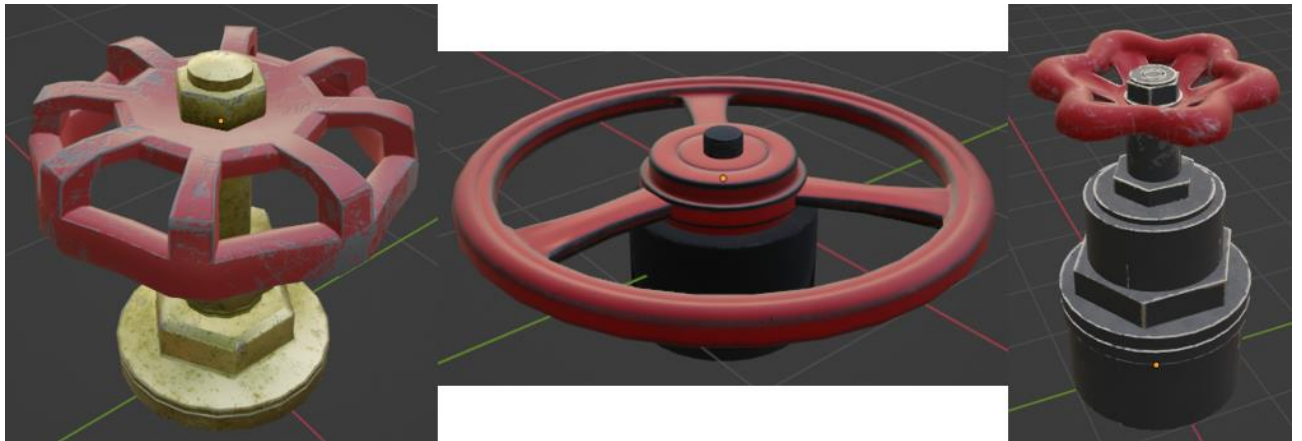


Рис.4.1. 3D-модели для интерактивного элемента «Вентиль»

### 4.2. Скрипт

#### Подготовка модели:

- 1) Необходимо наличие двух моделей: Одной моделью является основание элемента, а второй моделью — ручка элемента, с которой пользователь будет взаимодействовать.
- 2) Необходима настройка rigid body для обеих моделей: Необходимо установить режим rigid body (жесткое тело) для обеих моделей.
- 3) Необходима настройка позиции точки вращения – pivot point – для обеих моделей:
  - а. У основания элемента: Должна быть минимальной по оси y и находиться в середине по осям x и z.
  - б. У ручки элемента: Должна быть в центре вращения ручки.
- 4) Если объект требует вращения в процессе взаимодействия, необходимо настроить hinge joint: Для обеспечения возможности вращения объекта вокруг оси, в документации необходимо описать настройку hinge joint.

#### Добавление скрипта:

- 1) Создайте новый пустой объект или выберите существующий объект в сцене Unity.
- 2) Прикрепите скрипт "Valve" к выбранному объекту.
- 3) Установите необходимые значения для публичных переменных и настройте другие параметры по вашему выбору.

### **Настройка публичных переменных:**

- 1) myGameObject: Ссылка на игровой объект, который будет изменяться при использовании вентиля.
- 2) elementHandle: Ссылка на игровой объект, представляющий вентиль.
- 3) Valve Speed: Скорость поворота вентиля при удержании клавиши.
- 4) MinAngle: Минимальный угол поворота вентиля.
- 5) MaxAngle: Максимальный угол поворота вентиля.
- 6) ValveValue: Процентное значение поворота вентиля.
- 7) ValveOnMaxValue: Флаг, который указывает, находится ли вентиль в максимальном положении.

### **События:**

onValueChange: Событие, которое будет вызываться при изменении значения вентиля. Возвращает процент поворота вентиля в диапазоне от 0 до 1.

### **Режимы работы:**

- 1) changeModeButtonName: Название клавиши, которая будет использоваться для переключения режима работы (обычный режим или VR-режим).
- 2) isVRMode: Флаг, определяющий текущий режим работы вентиля.

### **Взаимодействие с вентилем посредством клавиатуры:**

Для использования вентиля используются клавиши, указанные в UseValveButtonName.

### **Взаимодействие с вентилем в VR-режиме:**

Вентиль может быть нажат путем захвата и перемещения контроллера в VR-режиме.

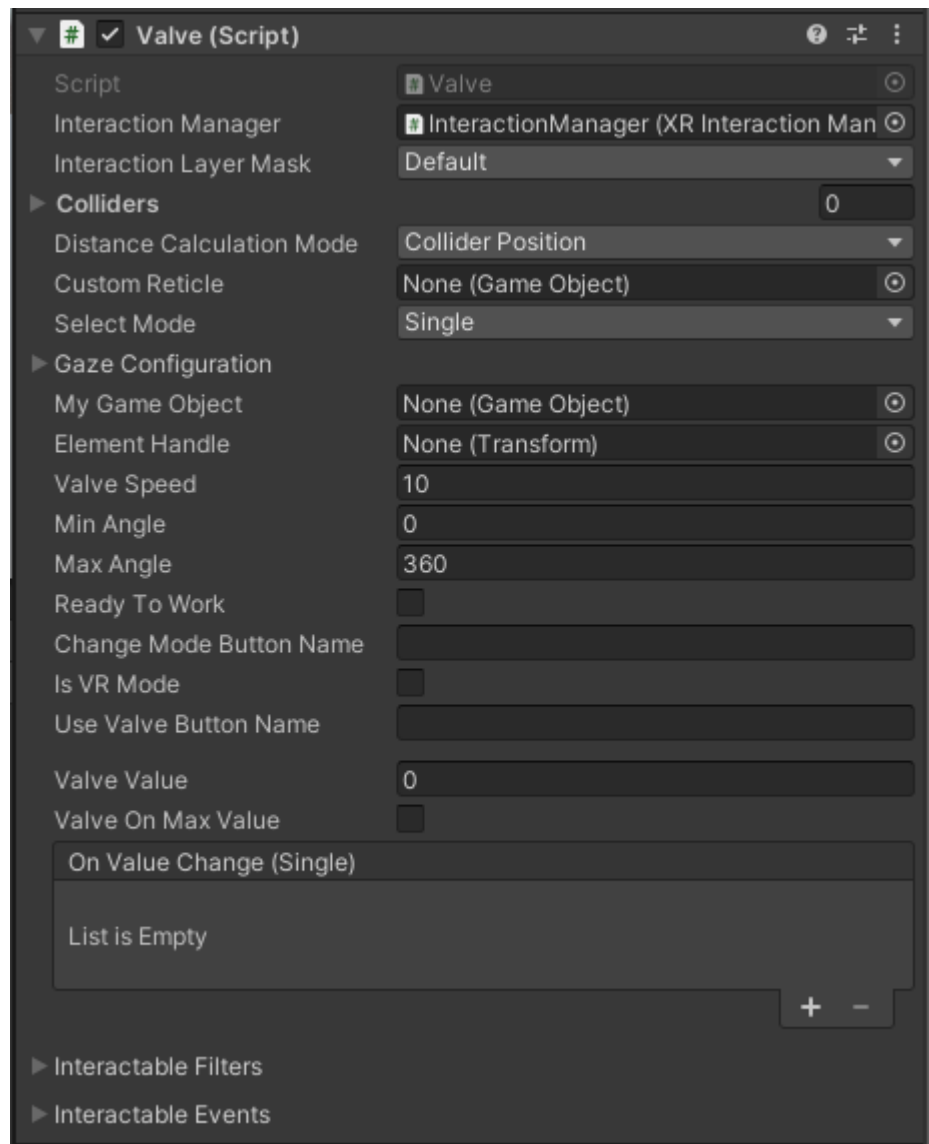


Рис.4.2. Скрипт Valve.cs в редакторе Unity

## 5. Ползунок

### 5.1. Модели

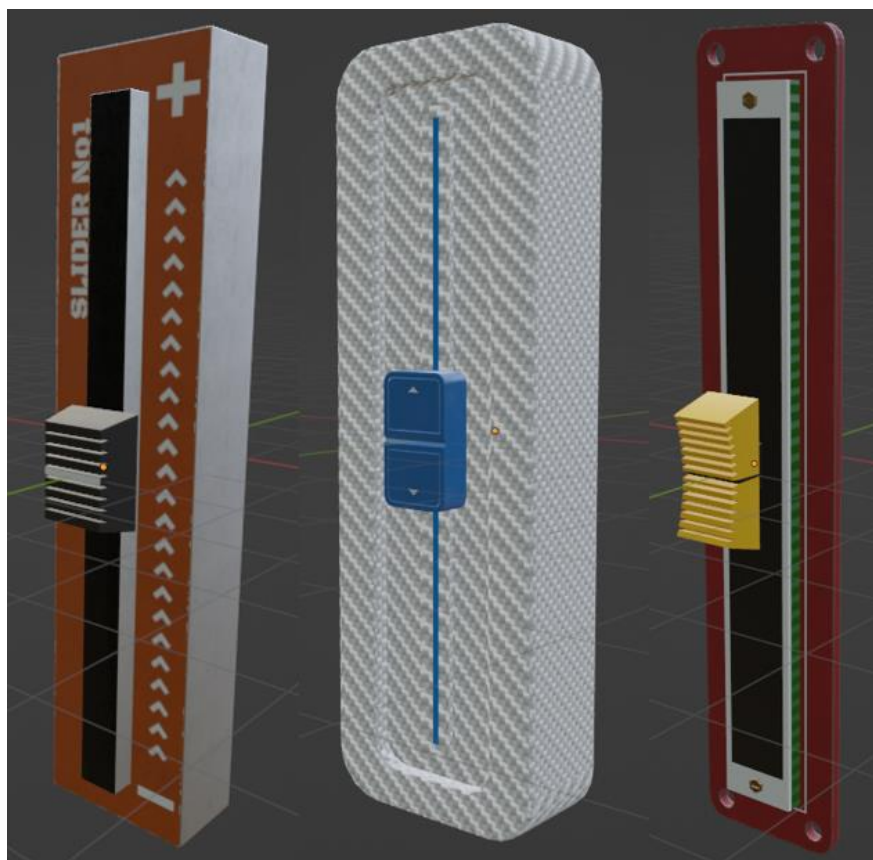


Рис.5.1. 3D-модели для интерактивного элемента «Ползунок»

### 5.2. Скрипт

#### Подготовка модели:

- 1) Необходимо наличие двух моделей: Одной моделью является основание элемента, а второй моделью — ручка элемента, с которой пользователь будет взаимодействовать.
- 2) Необходима настройка rigid body для обеих моделей: Необходимо установить режим rigid body (жесткое тело) для обеих моделей.
- 3) Необходима настройка позиции точки вращения – pivot point – для обеих моделей:
  - а. У основания элемента: Должна быть минимальной по оси y и находиться в середине по осям x и z.
  - б. У ручки элемента: Должна быть в центре объекта.

#### Добавление скрипта:

- 1) Создайте новый пустой объект или выберите существующий объект в сцене Unity.

- 2) Прикрепите скрипт "Slider" к выбранному объекту.
- 3) Установите необходимые значения для публичных переменных и настройте другие параметры по вашему выбору.

### **Настройка публичных переменных:**

- 1) myGameObject: Ссылка на игровой объект, который будет изменяться при использовании ползунка.
- 2) elementHandle: Ссылка на трансформ объекта, представляющего ползунок.
- 3) sliderUpperLimit: Верхний предел значения ползунка.
- 4) sliderLowerLimit: Нижний предел значения ползунка.
- 5) upButtonName: Название кнопки для увеличения значения ползунка в обычном режиме.
- 6) downButtonName: Название кнопки для уменьшения значения ползунка в обычном режиме.
- 7) segments: Количество сегментов ползунка.

### **События:**

onValueChange: Событие, которое будет вызываться при изменении значения ползунка. Возвращает текущее значение ползунка в диапазоне от 0 до 1.

### **Режимы работы:**

- 1) changeModeButtonName: Название клавиши, которая будет использоваться для переключения режима работы (обычный режим или VR-режим).
- 2) isVRMode: Флаг, определяющий текущий режим работы вентили.

### **Взаимодействие с ползунком посредством клавиатуры:**

Для увеличения и уменьшения значения ползунка используются кнопки, указанные в upButtonName и downButtonName.

### **Взаимодействие с ползунком в VR-режиме:**

Ползунок можно изменять, удерживая его виртуальным манипулятором и перемещая его вдоль оси X.



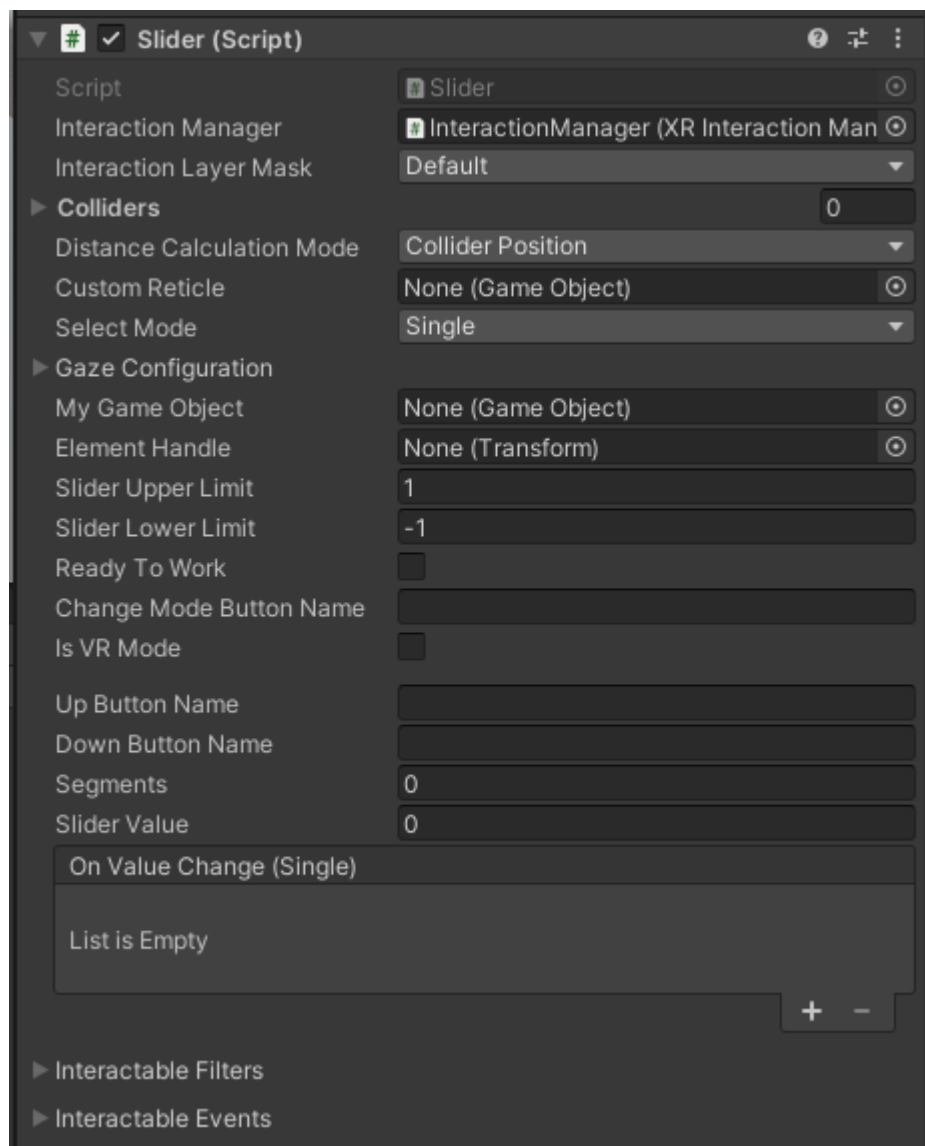


Рис.5.2. Скрипт Slider.cs в редакторе Unity

## 6. Переключатель

### 6.1. Модели

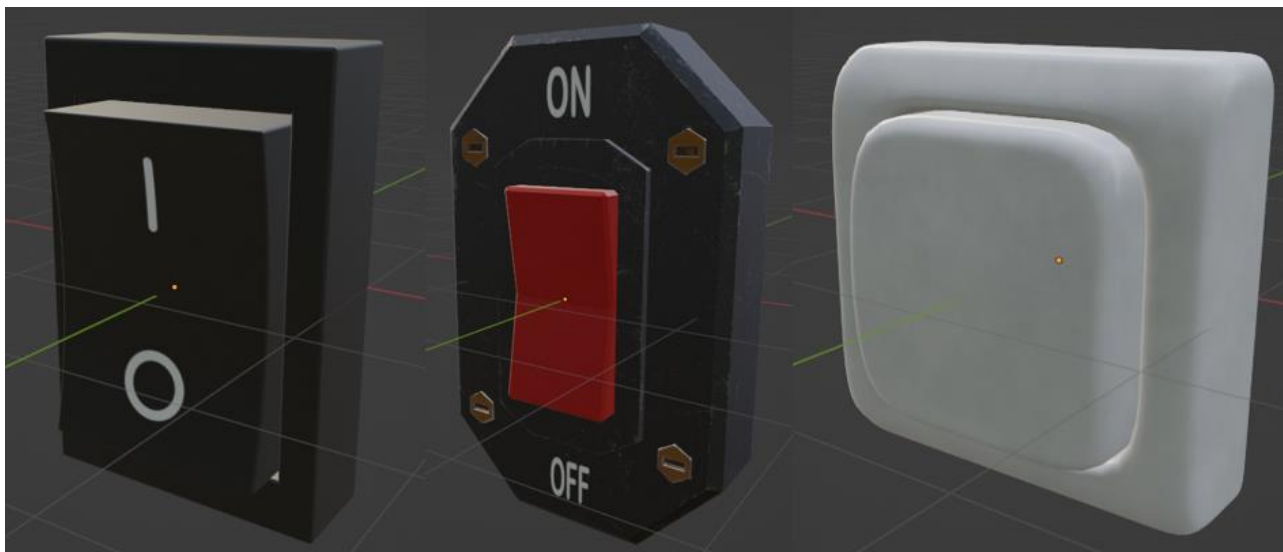


Рис.6.1. 3D-модели для интерактивного элемента «Переключатель»

### 6.2. Скрипт

#### Подготовка модели:

- 1) Необходимо наличие двух моделей: Одной моделью является основание элемента, а второй моделью — ручка элемента, с которой пользователь будет взаимодействовать.
- 2) Необходима настройка rigid body для обеих моделей: Необходимо установить режим rigid body (жесткое тело) для обеих моделей.
- 3) Необходима настройка позиции точки вращения – pivot point – для обеих моделей:
  - а. У основания элемента: Должна быть минимальной по оси y и находиться в середине по осям x и z.
  - б. У ручки элемента: Должна быть в центре вращения ручки.
- 4) Так как объект требует вращения в процессе взаимодействия, необходимо настроить hinge joint: Для обеспечения возможности вращения объекта вокруг оси, в документации необходимо описать настройку hinge joint.

#### Добавление скрипта:

- 1) Создайте новый пустой объект или выберите существующий объект в сцене Unity.
- 2) Прикрепите скрипт "Switch" к выбранному объекту.

3) Установите необходимые значения для публичных переменных и настройте другие параметры по вашему выбору.

### **Настройка публичных переменных:**

- 1) myGameObject: Ссылка на игровой объект, который будет изменяться при использовании переключателя.
- 2) elementHandle: Ссылка на игровой объект, представляющий ручку переключателя.
- 3) AngleMin: Минимальный угол поворота ручки переключателя.
- 4) AngleMax: Максимальный угол поворота ручки переключателя.

### **Режимы работы:**

- 1) changeModeButtonName: Название клавиши, которая будет использоваться для переключения режима работы (обычный режим или VR-режим).
- 2) isVRMode: Флаг, определяющий текущий режим работы вентили.

### **События:**

- 1) onPressedMin: Событие, которое будет вызываться при нажатии переключателя в минимальную позицию.
- 2) onPressedMax: Событие, которое будет вызываться при нажатии переключателя в максимальную позицию.

### **Взаимодействие с переключателем посредством клавиатуры:**

Используйте кнопку, указанную в pressSwitchName, чтобы нажать или отпустить переключатель.

### **Взаимодействие с кнопкой в VR-режиме:**

При перемещении VR-контроллера взглядом изменяется угол поворота переключателя.

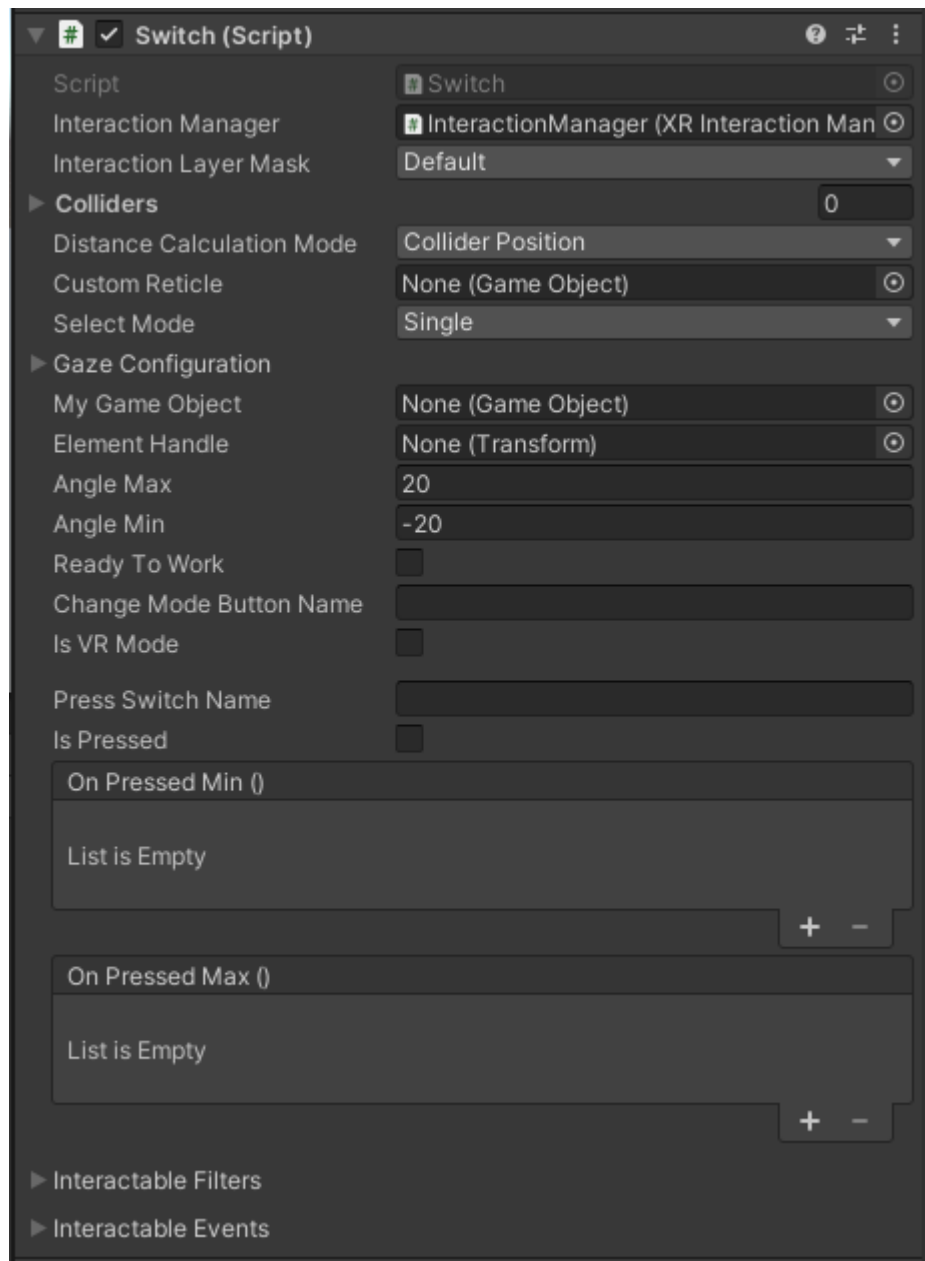


Рис.6.2. Скрипт Switch.cs в редакторе Unity

## 7. Визуальное подсвечивание

Визуальное подсвечивание кнопки будет включаться, если главная камера смотрит на кнопку из ближайшего расстояния.

### Добавление скрипта:

- 1) Создайте новый пустой объект или выберите существующий объект в сцене Unity.
- 2) Прикрепите скрипт "HighlightElements" к выбранному объекту.
- 3) Настройте список renderers, добавив все рендереры, которые требуется подсветить.

### Настройка публичных переменных:

- 1) renderers: Список рендереров, подлежащих подсветке.
- 2) colorKeyboard: Цвет подсветки для обычного режима.
- 3) colorVR: Цвет подсветки для VR-режима.

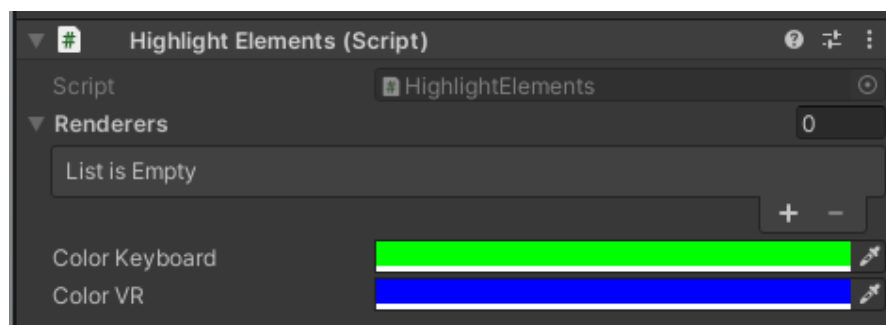


Рис.7.1. Скрипт HighlightElements.cs в редакторе Unity

### Пример работы:

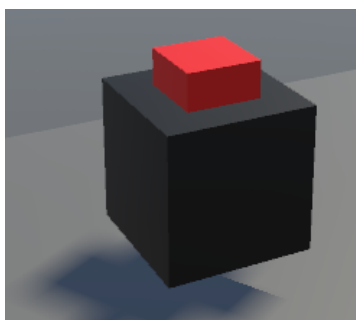


Рис.7.2. Неактивное состояние

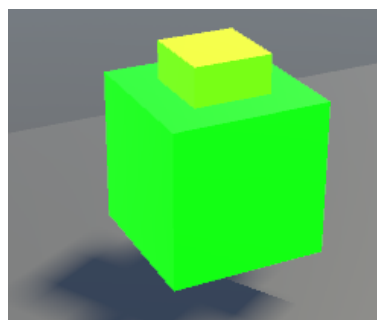


Рис.7.3. Активное состояние в режиме для клавиатуры

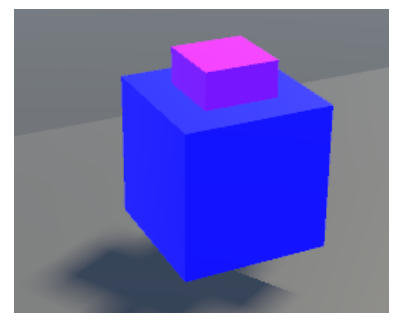


Рис.7.4. Активное состояние в режиме для VR