



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт информационных технологий (ИИТ)

Кафедра инструментального и прикладного программного обеспечения (ИиППО)

ОТЧЁТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ

Технологическая (проектно-технологическая) практика

приказ Университета о направлении на практику от «__» _____ 2023 г.
№ _____

Отчет представлен к
рассмотрению:

Студент группы ИКБО-20-19

«__» апреля 2023

Московка А.А.

(подпись и расшифровка подписи)

Отчет утвержден.
Допущен к защите:

Руководитель практики
от кафедры

«__» апреля 2023

Плотников С.Б.

(подпись и расшифровка подписи)

Москва 2023 г.



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт информационных технологий (ИИТ)

Кафедра инструментального и прикладного программного обеспечения (ИиППО)

ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ НА ПРОИЗВОДСТВЕННУЮ ПРАКТИКУ

Технологическая (проектно-технологическая) практика

Студенту 4 курса учебной группы ИКБО-20-19

Московке Артёму Александровичу

Место и время практики: РТУ МИРЭА кафедра ИиППО, с 23 марта 2023 г. по 19 апреля 2023 г.

Должность на практике: студент

1. СОДЕРЖАНИЕ ПРАКТИКИ:

1.1. Изучить:

1.1.1. Передовую научную, методологическую и инженерную литературу, включая научные статьи, диссертации, монографии, отчеты;

1.1.2. Современные методы моделирования, анализа и использования формальных методов конструирования программного обеспечения.

1.2. Практически выполнить:

1.2.1 Сформулировать целеполагание практики;

1.2.2. Построить блок-схемы логических сценариев виртуальных элементов симуляции и взаимодействия пользователя с ними;

1.2.3. Разработать логические сценарии виртуальных элементов симуляции.

1.3. Ознакомиться:

1.3.1. С национальными и международными стандартами, определяющими методологии работ в выбранной предметной области.

1.3.2. С компетенциями, реализуемыми в процессе выполнения работ по программе практики.

2. ДОПОЛНИТЕЛЬНОЕ ЗАДАНИЕ: подготовить доклад на научно-техническую конференцию студентов и аспирантов РТУ МИРЭА или иную конференцию, подготовить презентационный материал

3. ОРГАНИЗАЦИОННО-МЕТОДИЧЕСКИЕ УКАЗАНИЯ: В процессе практики рекомендуется использовать периодические издания и отраслевую литературу годом издания не старше 5 лет от даты начала прохождения практики

Руководитель практики от кафедры
«23» марта 2023 г.

Подпись (Плотников С.Б.)

Задание получил
«23» марта 2023 г.

Подпись (Московка А.А.)

СОГЛАСОВАНО:

Заведующий кафедрой:

«23» марта 2023 г.

Подпись (Болбаков Р.Г.)

Проведенные инструктажи:

Охрана труда:

Инструктирующий

Подпись

«23» марта 2023 г.

Болбаков Р.Г., зав. кафедрой
ИиППО

Инструктируемый

Подпись

Московка А.А.

Техника безопасности:

Инструктирующий

Подпись

«23» марта 2023 г.

Болбаков Р.Г., зав. кафедрой
ИиППО

Инструктируемый

Подпись

Московка А.А.

Пожарная безопасность:

Инструктирующий

Подпись

«23» марта 2023 г.

Болбаков Р.Г., зав. кафедрой
ИиППО

Инструктируемый

Подпись

Московка А.А.

С правилами внутреннего распорядка ознакомлен:

Подпись

«23» марта 2023 г.

Московка А.А.



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

РАБОЧИЙ ГРАФИК ПРОВЕДЕНИЯ ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ

студента Московки А.А. 4 курса группы ИКБО-20-19 очной формы обучения, обучающегося по направлению подготовки 09.03.04 Программная инженерия, профиль «Разработка программных продуктов и проектирование информационных систем».

| Неделя | Сроки выполнения | Этап | Отметка о выполнении |
|--------|-----------------------|---|----------------------|
| 1 | 23.03.2023 | Подготовительный этап, включающий в себя организационное собрание (Вводная лекция о порядке организации и прохождения производственной практики, инструктаж по технике безопасности, получение задания на практику) | |
| 1-4 | 23.03.2023-13.04.2023 | Изучение передовой научной, методологической и инженерной литературы, современные методы моделирования, анализа и использования формальных методов конструирования ПО; практическое выполнение: формулирование целеполагания практики, построение блок-схемы логических сценариев виртуальных элементов симуляции и взаимодействия пользователя с ними, разработка логических сценариев виртуальных элементов симуляции | |
| 4 | 13.04.2023 | Представление руководителю 3 главы отчета по практике | |
| 4 | 13.04.2023-18.04.2023 | Подготовка окончательной версии отчета по практике (Оформление материалов отчета в полном соответствии с требованиями на оформление письменных учебных работ студентов) | |
| 4 | 19.04.2023 | Представление окончательной версии отчета по практике руководителю | |

Руководитель практики от
кафедры

_____/Плотников С.Б., к.т.н., доцент/

Обучающийся

_____/Московка А.А./

Согласовано:

Заведующий кафедрой

_____/Болбаков Р.Г., к.т.н., доцент/

РЕФЕРАТ

Отчет 42 с., 27 рис., 7 источн.

СИМУЛЯЦИЯ, ЛОГИЧЕСКИЙ СЦЕНАРИЙ, ВИРТУАЛЬНЫЙ ЭЛЕМЕНТ, ХИМИЧЕСКАЯ ЛАБОРАТОРИЯ, СТЕРИЛЬНАЯ КОМНАТА

Объект исследования – симуляция химической лаборатории для инструктирования и тестирования обучающихся правилам взаимодействия с ней.

Предмет исследования – обучающее симуляционное ПО, имитирующее процессы и поведение специалиста в химической лаборатории.

Целью работы является изучение научной и инженерной литературы по выбранной тематике, формулирование целеполагание практики, построение блок-схем логических сценариев виртуальных элементов симуляции и взаимодействия пользователя с ними, разработка непосредственно логических сценариев виртуальных элементов симуляции.

Методом сравнительного анализа были определены перспективные решения и реализация требуемых логических сценариев трехмерных клонов, включая визуальное отображение скриптов, симулируемые процессы и варианты поведения пользователя в виртуальной среде. Смоделированы удобные для восприятия графические схемы логических сценариев, наглядно демонстрирующие всевозможные варианты развития событий при взаимодействии пользователя с элементами окружения в разрабатываемой симуляции. Наконец, разработаны сами скрипты поведения трехмерных клонов, которые будут использоваться в создаваемой фотореалистичной симуляции стерильной комнаты.

Результатом работы является набор блок-схем логических сценариев виртуальных элементов симуляции и сами логические сценарии, готовые к последующей интеграции в разрабатываемую виртуальную лабораторию.

СОДЕРЖАНИЕ

| | |
|---|----|
| ВВЕДЕНИЕ | 9 |
| 1 Построение блок-схем логических сценариев виртуальных элементов симуляции и взаимодействия пользователя с ними | 10 |
| 2 Разработка логических сценариев виртуальных элементов симуляции | 37 |
| ЗАКЛЮЧЕНИЕ | 41 |
| СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ | 42 |

ВВЕДЕНИЕ

Поскольку разработка симуляционного ПО производится на основе игрового движка Unreal Engine 5, в рамках данной практической работы будут неоднократно упомянуты такие присущие выбранной среде разработки элементы, как блюпринт, событие, компонент, задержка, таймлайн [1].

Блюпринтом в Unreal Engine называется особый программный элемент, представляющий собой набор каких-то конкретных компонентов, программную логику в виде узлов, связанных между собой, а также переменные и данные, которые хранятся в этих переменных и используются для создания требуемой логики выполнения последовательности [2].

Событием называются узлы блюпринта, которые вызываются из кода основной программы, которые начинают выполнение определенной последовательности операций, определенной разработчиком [3].

Компонентом является особый вид класса Object ядра Unreal Engine, который может представлять собой определенное поведение или являться каким-то трехмерным объектом, а именно статической сеткой объекта [4].

Задержкой называется специальный узел, буквально устанавливающий необходимое время ожидания перед продолжением выполнения последовательности скрипта [5].

Таймлайном называется график функции зависимости определенной переменной от времени [6].

Для более наглядного и понятного представления блюпринтов в первой главе будут спроектированы блок-схемы логических сценариев с подробным описанием назначения каждого элемента и четко определенной последовательностью действий. Во второй главе будут разработаны, описаны и представлены сами логические сценарии с детальным объяснением назначения каждого элемента.

1 Построение блок-схем логических сценариев виртуальных элементов симуляции и взаимодействия пользователя с ними

Все представленные далее блок-схемы в данной главе и соответствующие логические сценарии в следующей главе расположены в порядке увеличения объема с целью равномерного и плавного повышения сложности восприятия логических сценариев и по фактору наличия связей между несколькими логическими сценариями, усложняющими восприятие.

Блок-схемы идут в строгой последовательности в направлении от верхнего к нижнему блоку слева направо, у каждой схемы имеется блок начала и конца, символизирующие соответственно начало и конец последовательности логического сценария.

По причине того, что для представления логических сценариев не предусмотрено стандартов по оформлению в виде блок-схем, будет использована следующая система обозначений: прямоугольным блоком будет являться вызов функции, прямоугольным блоком с вертикальными линиями по краям будет являться вызов события из программы, ромбом будет обозначаться логическое ветвление. Также могут присутствовать дополнительные компоненты и переменные, обозначенные блоками с закругленными краями.

Некоторые блюпринты имеют несколько событий, поэтому для них будет представлено более одной блок-схемы.

Блок-схемы составлялись из соображений логики и структуры работы игрового ядра Unreal Engine.

Самыми простыми скриптами были выбраны скрипты смены материала трехмерных моделей ультрафиолетовых ламп с материала выключенной лампы на материал включенной лампы по вызову определенного события посредством взаимодействия с настенным выключателем, который реализован блюпринтом WallSwitchBP. На рисунках 1.1-1.2 представлены блок-схемы логических

сценариев событий включения и выключения УФ-лампы turnOnUVLamp и turnOffUVLamp соответственно.

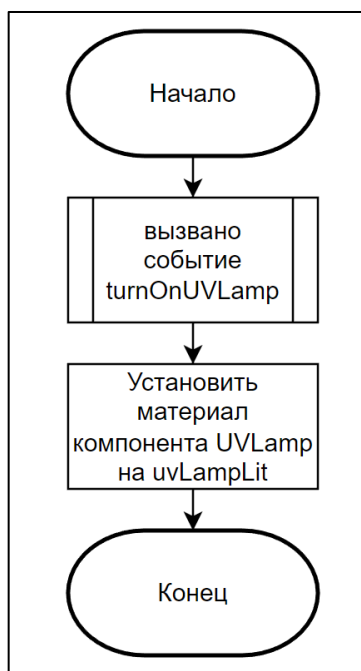


Рисунок 1.1 – Скриншот блок-схемы события turnOnUVLamp

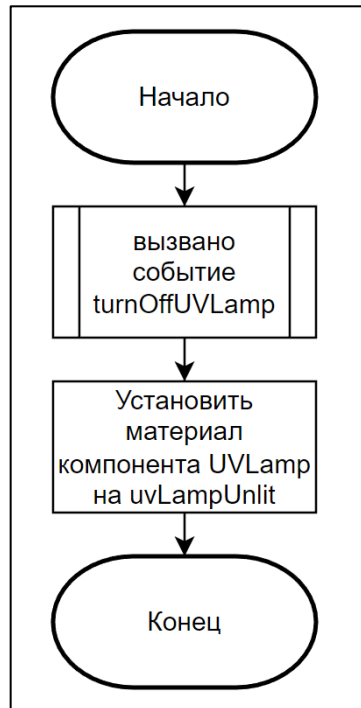


Рисунок 1.2 – Скриншот блок-схемы события turnOffUVLamp

Следующими по сложности выступают скрипты взаимодействия манометров при открытии герметичных дверей и тем самым появлением нестабильного давления в отсеках лаборатории.

На рисунке 1.3 изображена блок-схема связи событий запуска анимации движения стрелки манометра и ее остановки при вызове таковых открытием дверей: для создания графика зависимости функции от времени использовался таймлайн, изображенный на схеме в виде цикла, условием завершения которого является вызов события остановки манометра. В теле цикла происходит изменение переменной `ArrowRotation`, отвечающей за угол вращения стрелки манометра.

Скриншот используемого в данном таймлайне графика функции переменной `ArrowRotation` можно увидеть на рисунке 2.1 (см. вторую главу).

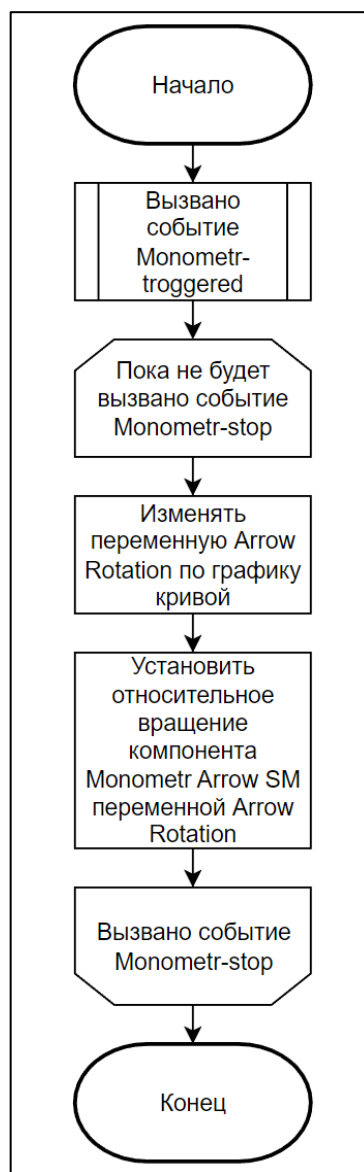


Рисунок 1.3 – Скриншот блок-схемы связи событий Monometr-troggered и Monometr-stop

При закрытии герметичных шлюзов и стабилизации давления в отсеках вызывается событие Monometr-onClosedDoor, представленное далее на рисунке 1.4. Цикл анимации возврата стрелки манометра в начальное положение протекает до тех пор, пока не закончится график кривой, на котором изображено плавное изменение переменной ArrowStoppingRotation из нестабильного в стабильное положение, равные примерно 13 и 10 паскаль соответственно.

График функции переменной ArrowStoppingRotation для используемого в данном скрипте таймлайна представлен далее на рисунке 2.2 (см. вторую главу).

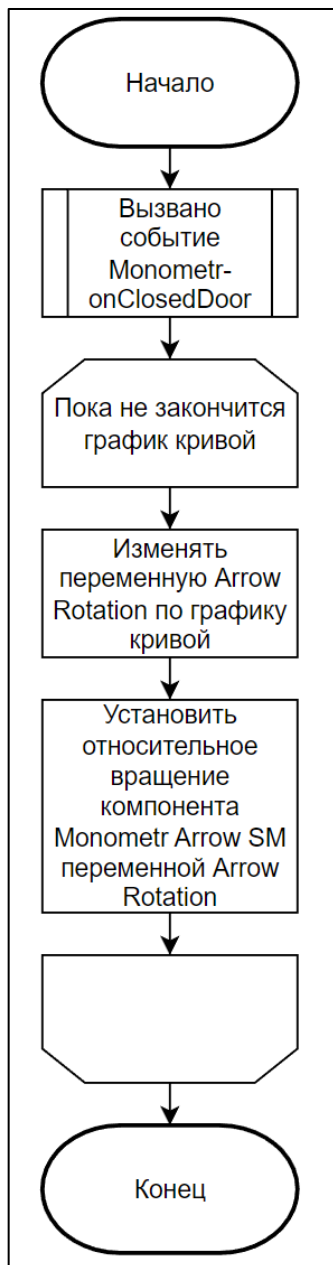


Рисунок 1.4 – Скриншот блок-схемы события Monometr-onClosedDoor

Далее была построена блок-схема блюпринта настенного включателя ультрафиолетовой лампы, выполняющей стерилизацию рабочего помещения. УФ-лампу возможно включить только при условии, что не нарушена герметичность и в помещении никого нет, поэтому сразу по вызову события

Event Interact с настенным включателем происходит проверка условия, открыта ли вторая дверь, преграждающая путь из комнаты с включателем УФ-лампы в основную рабочую зону с самой лампой. В случае, если дверь открыта и нарушена герметичность помещения, на экран выводится на две секунды текстовое сообщение, объясняющее правило взаимодействия с УФ-лампой при открытых дверях. Первую часть скрипта можно видеть на рисунке 1.5.

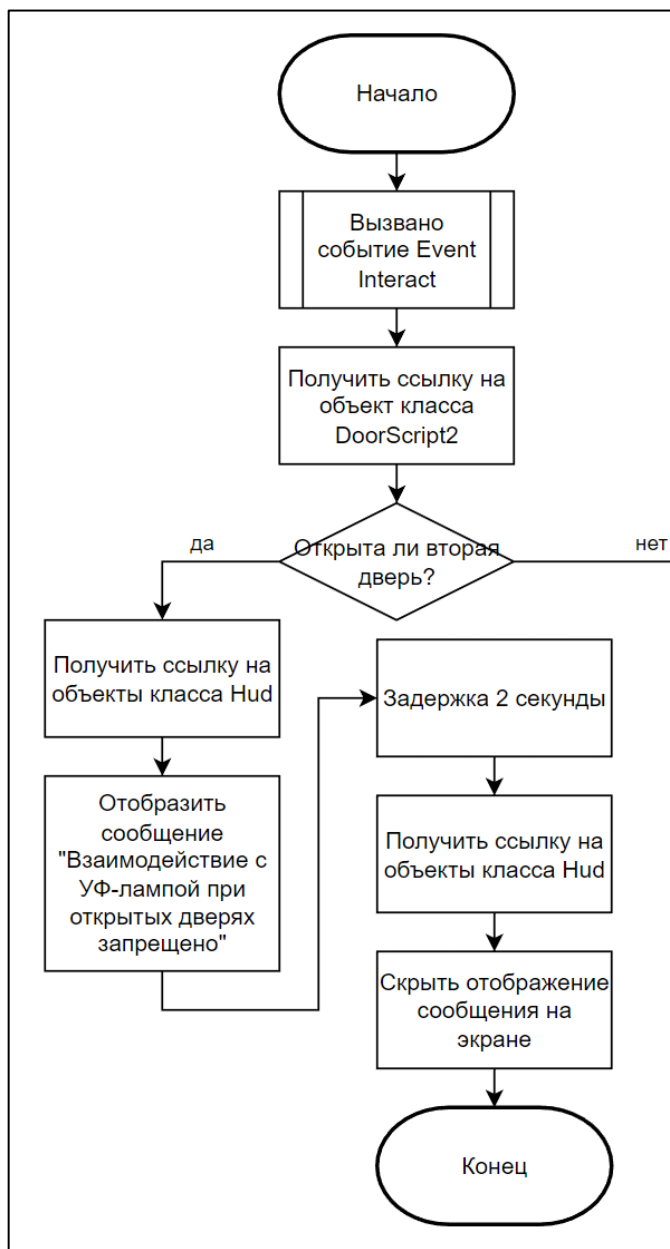


Рисунок 1.5 – Скриншот первой части блок-схемы скрипта включателя УФ-лампы

В ином случае после получения ссылки на объект класса UVLightBP происходит проверка, включена ли УФ-лампа или нет. Если лампа включена, соответствующая данному условию булева переменная `isOn` устанавливается в состояние `False`, после чего вызывается событие выключения УФ-лампы `turnOffUVLamp`, после которого на экран на две секунды выводится сообщение о том, что УФ-лампа выключена, после чего граф операций завершается. Если же лампа выключена, происходят обратные действия, а именно: булева переменная состояния работы лампы `isOn` устанавливается на `True`, вызывается событие включения лампы `turnOnUVLamp`, наконец, на две секунды на экран выводится информационное сообщение о том, что УФ-лампа включена, после чего скрипт завершает свое действие. Вторая часть блок-диаграммы представлена на следующем рисунке 1.6 ниже.

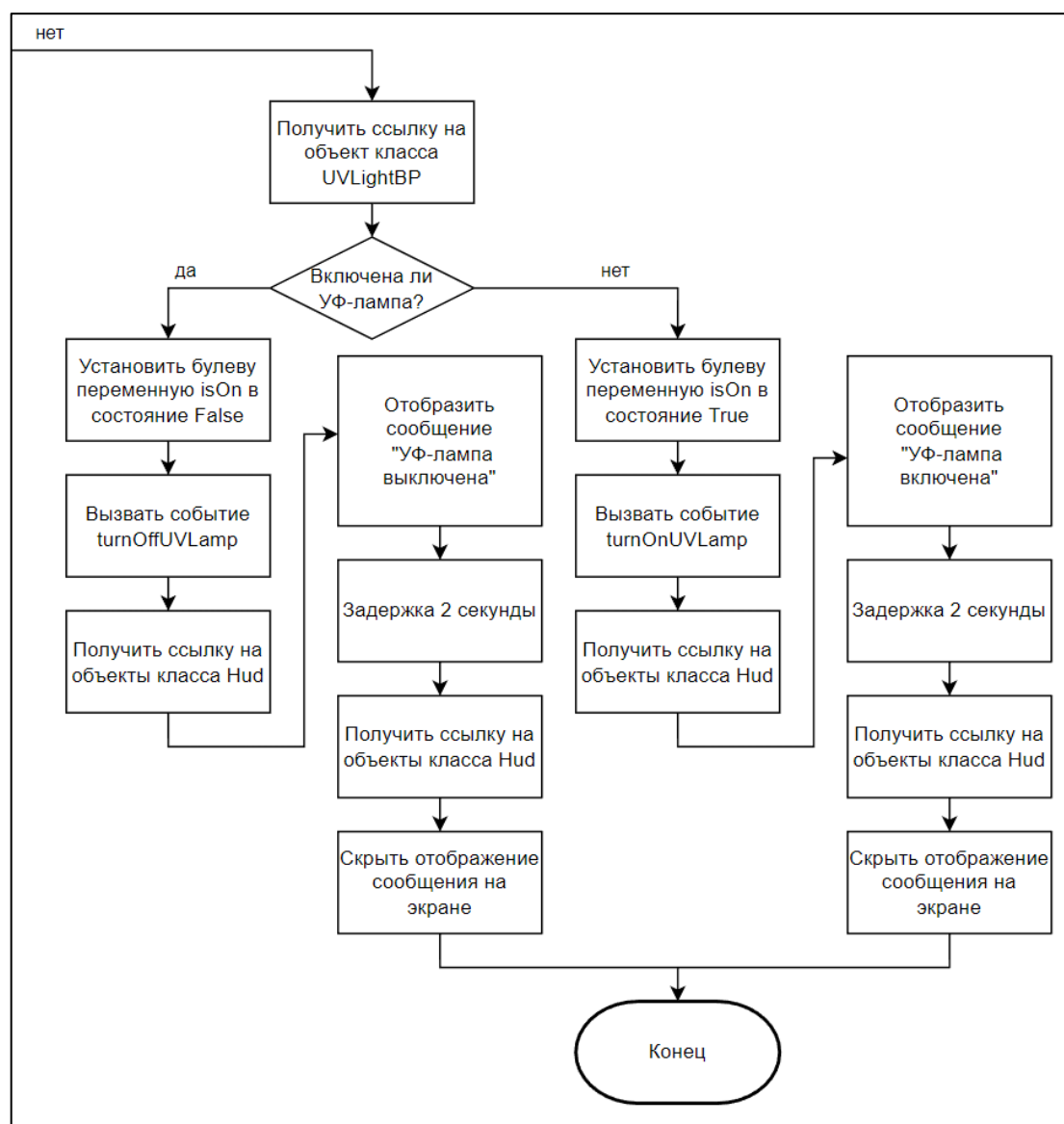


Рисунок 1.6– Скриншот второй части блок-схемы скрипта включателя УФ-лампы

Следующими по сложности и очередности были рассмотрены блюпринты первых двух герметичных дверей между входным помещением, комнатой переодевания перед рабочей зоной и основной рабочей зоной лаборатории. Остальные две двери не будут затронуты в рамках данной практической работы в силу своей идентичности рассмотренным далее первой и второй дверям.

Сначала была построена блок-схема первой двери, особенностью которой является специальная булева переменная `lock`, контролирующая, осмотрелся ли пользователь в начальном помещении перед тем, как продвигаться далее.

После взаимодействия с первой дверью происходят проверки условия, открыты ли соседние от первой четвертая и вторая двери, находящиеся в начальной комнате и комнате переодевания соответственно. В случае, если какая-то из дверей открыта, на две секунды выводится информационное сообщение на экран с просьбой закрыть эту дверь для восстановления стабильного уровня давления и восстановления герметичности в помещении. Данную часть логики можно увидеть на рисунке 1.7 далее.

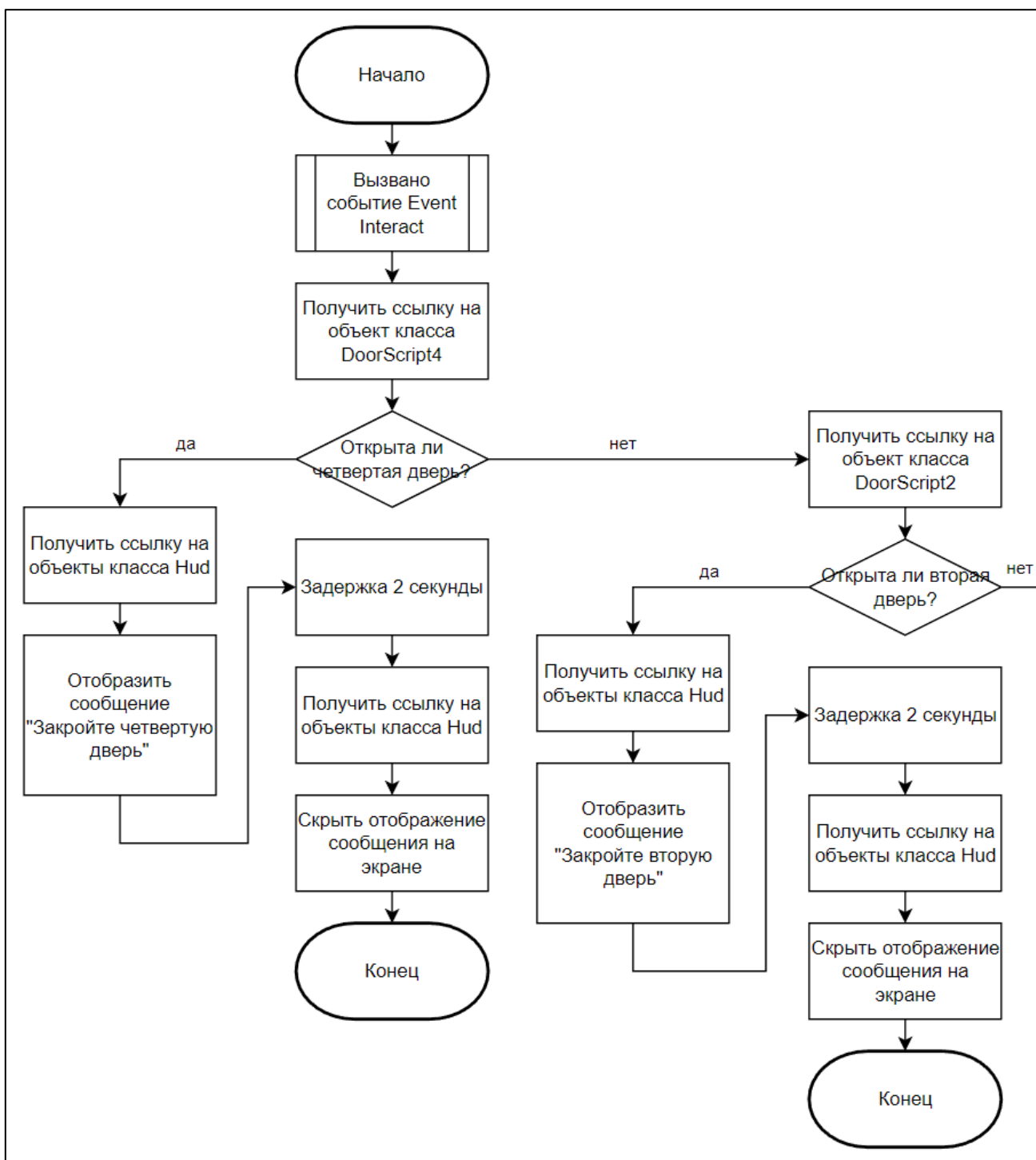


Рисунок 1.7 – Скриншот первой части блок-схемы скрипта первой двери

В случае же, если соседние двери закрыты, происходит проверка блокировки двери перед ее открытием: если пользователь не осмотрелся в первой комнате, в консоль выводится сообщение “Step further”. В ином случае

последующая проверка уточняет, закрыта ли эта дверь: если дверь закрыта, происходит поиск всех объектов класса MonometrBP с последующим вызовом событий Monometr-troggered (это выполняется по той причине, что экземпляров класса MonometrBP на сцене несколько), после чего запускается звуковая дорожка открытия двери и изменяется тип коллизии двери на “BlockDoors” с целью отключения возможности взаимодействия с дверью во время ее анимации закрытия или открытия, при котором можно было бы нарушать и наслаивать звуковые дорожки открытия дверей. После этого в цикле посредством таймлайна происходит плавная анимация открытия двери с использованием графика функции. После этого цикла изменяется тип коллизии двери на изначальный с целью возврата возможности взаимодействия пользователя с дверью для ее закрытия. Если же дверь была изначально открыта, то происходят противоположные описанным ранее действия: проигрывается звуковая дорожка закрытия двери, отключается возможность прервать анимацию закрытия двери, в цикле с использованием таймлайна того же графика в обратном направлении движения функции производится плавная анимация закрытия двери, после чего пользователю возвращается возможность взаимодействия с дверью. Все эти действия изображены на рисунке 1.8.

График функции DoorRotationZ, используемой в таймлайне, изображен на рисунке 2.3 (см. вторую главу).

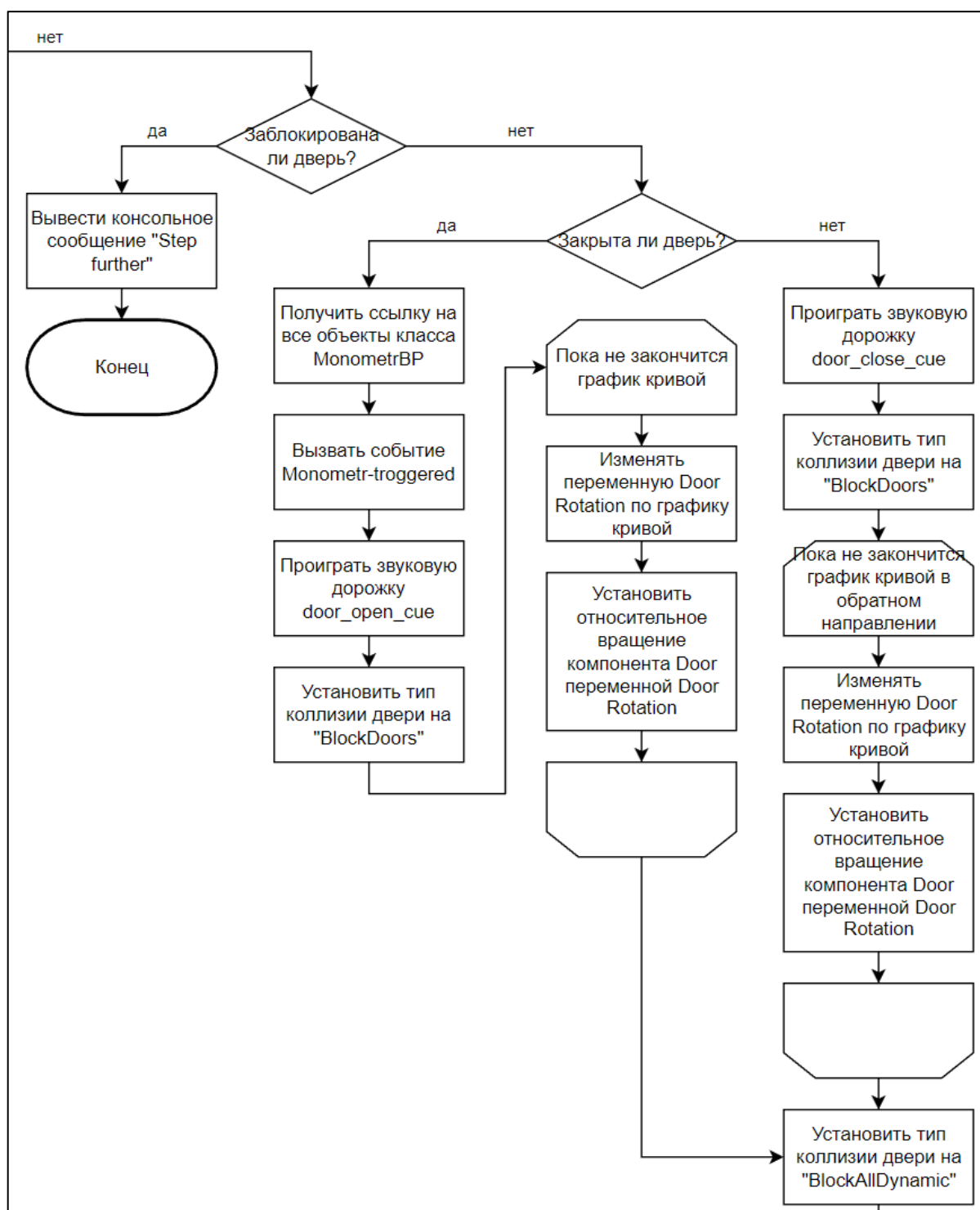


Рисунок 1.8 – Скриншот второй части блок-схемы скрипта первой двери

На последнем рисунке 1.9 данного блюпринта изображено завершающее условие, проверяющее, закрыта ли дверь: если дверь закрыта, то граф операций завершается, в ином случае происходит получение всех ссылок на объекты класса

MonometrBP, по которым вызываются события остановки анимации стрелки манометров и последующего возвращения стрелок этих манометров в стабильное положение.

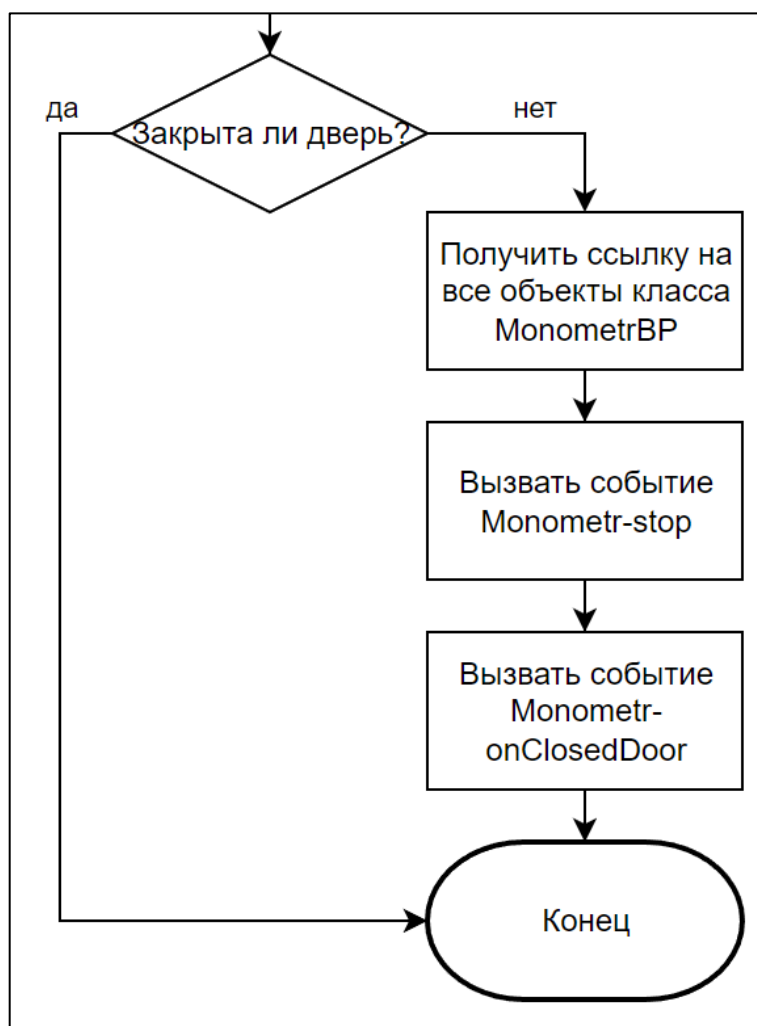


Рисунок 1.9 – Скриншот третьей части блок-схемы скрипта первой двери

Далее была построена блок-схема второй двери, особенностью которой является логика, проверяющая состояние работы ультрафиолетовой лампы в основном помещении, перед которым эта дверь и расположена.

На рисунке 1.10 изображена первая часть бюлпринта второй двери, в которой после взаимодействия с дверью происходит проверка состояния соседних первой и третьей дверей, в случае открытого состояния которых будет

выведено на экран на две секунды сообщение с просьбой закрыть эти двери для устранения разгерметизации.

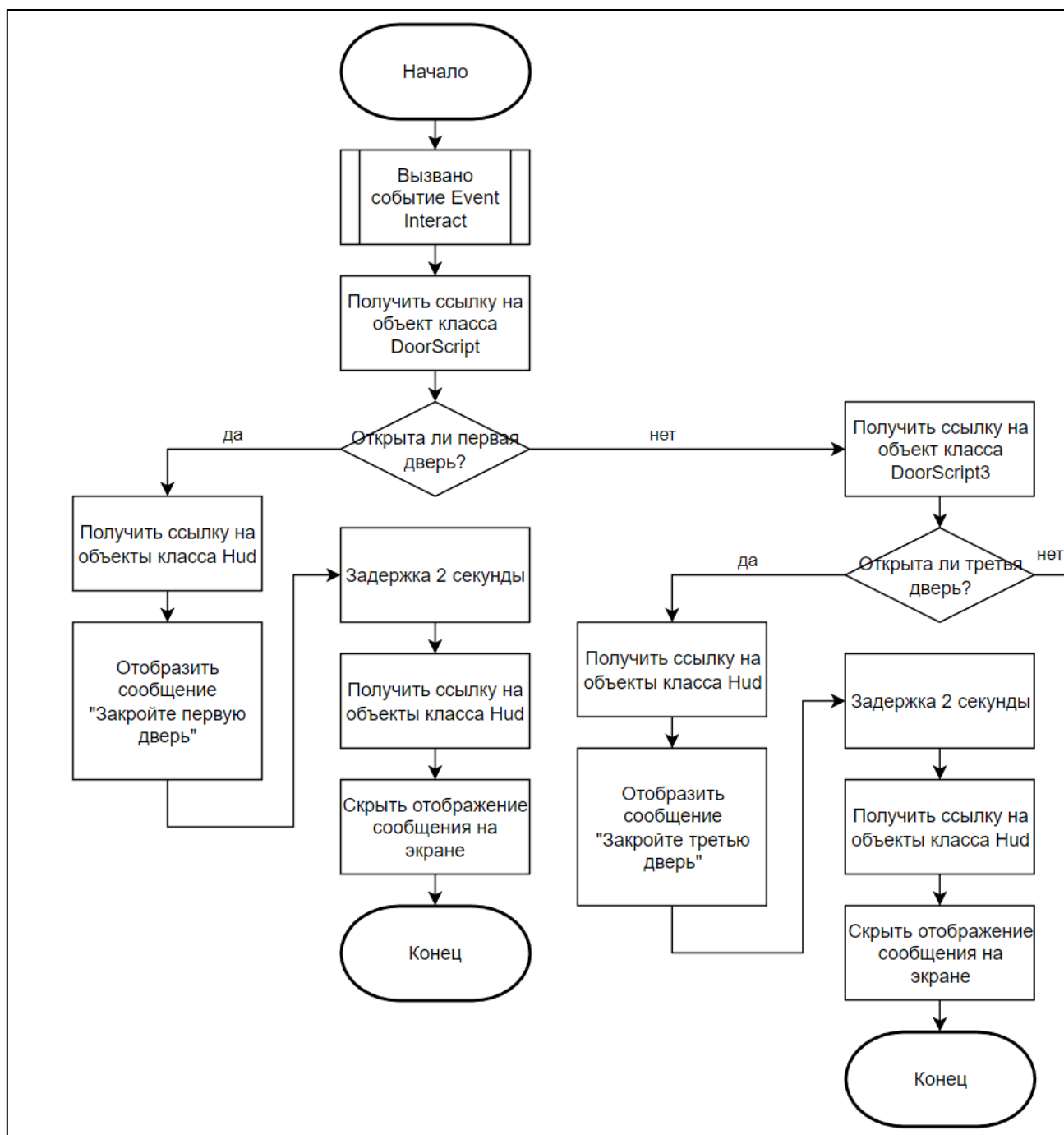


Рисунок 1.10 – Скриншот первой части блок-схемы скрипта второй двери

На следующем рисунке 1.11 изображена вторая часть схемы, в которой проверяется состояние ультрафиолетовой лампы, в случае работы которой

открыть дверь не предоставляется возможным и выводится соответствующее информационное сообщение. В случае же выключенного состояния УФ-лампы возможность открыть дверь предоставляется и в зависимости от ее актуального состояния будет выбрана последовательность действий. Если дверь закрыта, то производится получение ссылок на все объекты класса MonometrBP с целью вызова принадлежащего ему события Monometr-troggered, после которого запускается воспроизведение звуковой дорожки открытия двери, а возможность прервать последующую анимацию открытия двери отключается. Далее в цикле благодаря графику кривой в таймлайне (см. рисунок 2.3 второй главы) происходит плавное изменение относительного угла вращения двери, после успешного выполнения анимации движения которой пользователю возвращается возможность взаимодействия с дверью. Если вернуться к ветвлению выше в случае открытой двери, первым делом проигрывается звуковая дорожка закрытия двери, затем отключается возможность прервать анимацию закрытия двери, в цикле посредством вышеупомянутого графика кривой, но в обратном направлении изменения функции происходит плавное изменение относительного угла вращения двери, после которого дверь успешно закрывается и пользователю возвращается возможность взаимодействия с дверью.

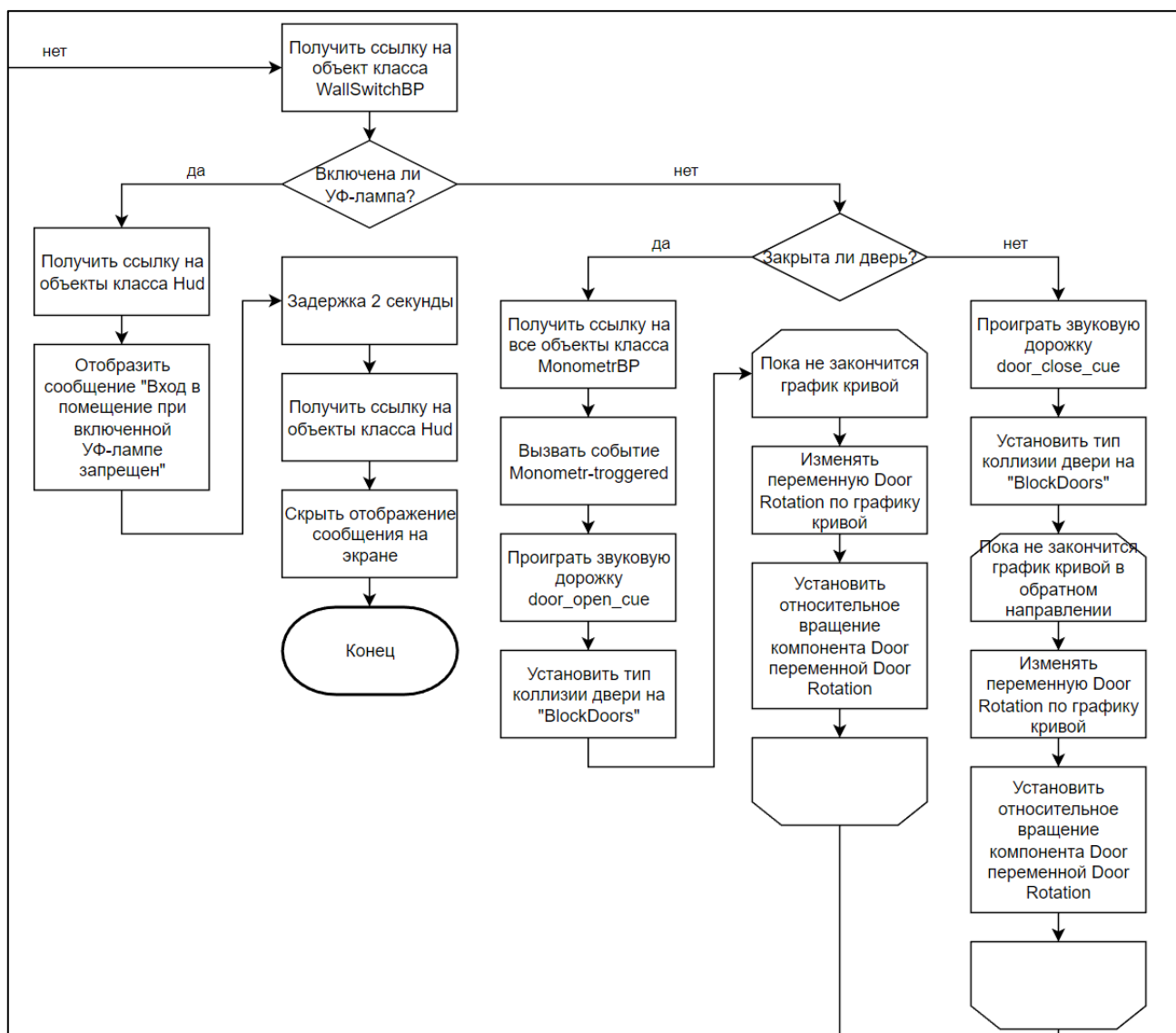


Рисунок 1.11 – Скриншот второй части блок-схемы скрипта второй двери

Наконец, на рисунке 1.12 происходит последняя проверка состояния двери, в зависимости от результатов которой либо не происходит ничего в случае закрытой двери, либо происходит получение ссылок на все объекты класса MonometrBP, посредством которых вызываются два события: Monometr-stop, останавливающий анимацию колебания стрелки манометра, а также Monometr-onClosedDoor, возвращающий стрелку манометра в стабильное состояние нуля паскаль.

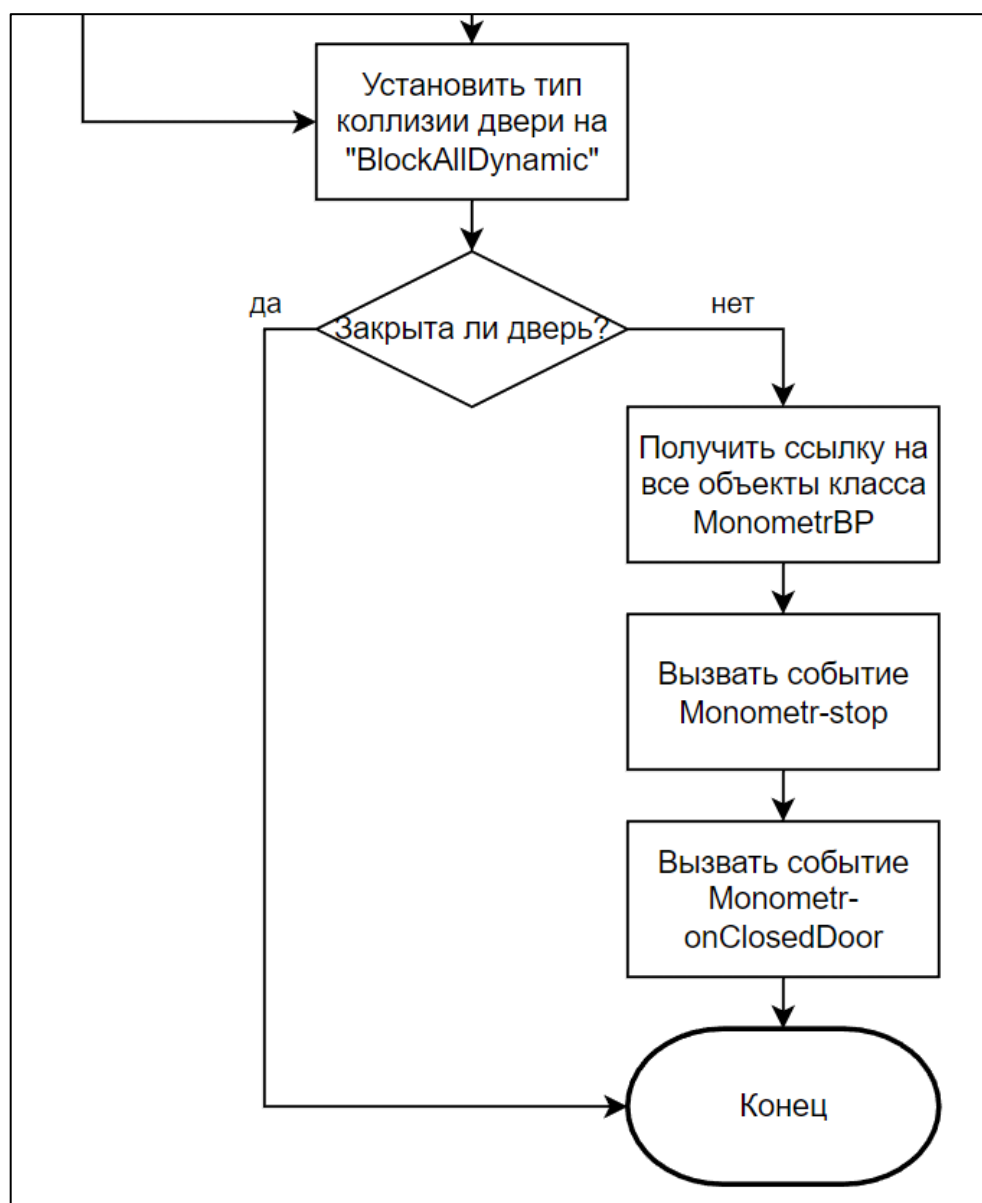


Рисунок 1.12 – Скриншот третьей части блок-схемы скрипта второй двери

Последним в рамках первой главы будет рассмотрен блюпринт герметичного шлюза передачи вещей из нестерильного помещения в стерильное. Для удобства просмотра и анализа блок-схема была разделена на девять частей, которые будут представлены на рисунках 1.13-1.21.

При взаимодействии с шлюзом передачи вещей первостепенно сохраняется ссылка на компонент, с которым было произведено взаимодействие, в отдельную переменную для дальнейшего взаимодействия с ней. Затем производится

сравнение того, с каким объектом было произведено взаимодействие, с тем, какой из компонентов реализует какую из четырех ветвей взаимодействия. В специальную целочисленную переменную switch сохраняется значение, присвоенное определенному компоненту: если взаимодействие произошло с внешней дверью, переменной присваивается 0, если взаимодействие произошло с внутренней дверью – присваивается 1, если же взаимодействие произошло с внешней кнопкой – присваивается 2, наконец, если взаимодействие произошло с внутренней кнопкой – присваивается 3. От этой переменной switch в логическом узле происходит ветвление потока данных аналогичным условным оператором switch. Данное поведение изображено на рисунке 1.13 ниже.

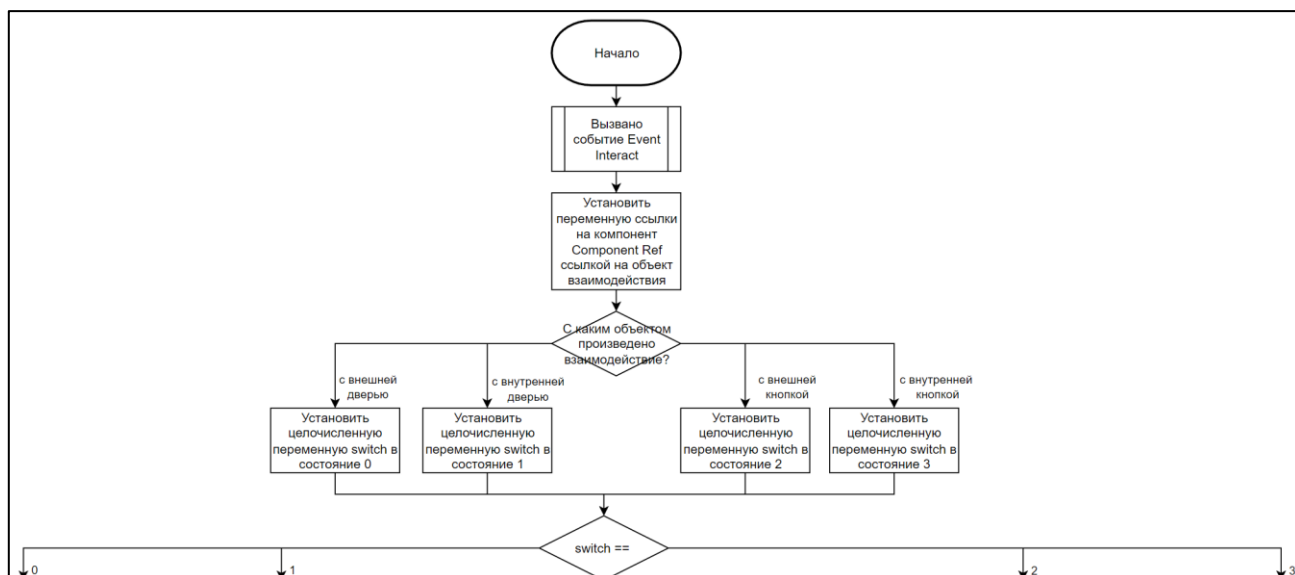


Рисунок 1.13 – Скриншот первой части блок-схемы скрипта шлюза передачи вещей

На рисунке 1.14 изображена первая половина графа операций, связанного с взаимодействием с внешней дверью шлюза передачи предметов.

Сначала проверяется условие: включена ли УФ-лампа – если лампа включена, то уже неоднократно описанным ранее образом производится вывод на экран информационного сообщения с извещением пользователя о том, что во время работы УФ-лампы открытие шлюза запрещено. Если же лампа выключена,

следующий логический узел проверяет, открыта ли внутренняя дверь (являющаяся внешней по отношению к данной двери, с которой было произведено взаимодействие). Если внутренняя дверь открыта, на экран на две секунды выводится сообщение, призывающее закрыть обратный шлюз. В ином случае соответствующей булевой переменной `outDoorIsOpen` присваивается значение, противоположное стоящему ранее. Затем производится проверка условия, открыта ли внешняя дверь по отношению к той, с которой взаимодействовали. Если дверь открыта, начинается процесс закрытия двери шлюза для передачи вещей: устанавливается световой индикатор открытого шлюза красного цвета, затем отключается возможность взаимодействия с внешними дверью и кнопкой. Затем в цикле производится плавное изменение относительного угла вращения дверцы шлюза до полного ее закрытия посредством графика функции в таймлайне, после чего пользователю возвращается возможность взаимодействия с внешними элементами шлюза передачи вещей.

Используемые в данном блюпринте графики функций будут изображены на рисунках 2.4-2.5 (см. вторую главу).

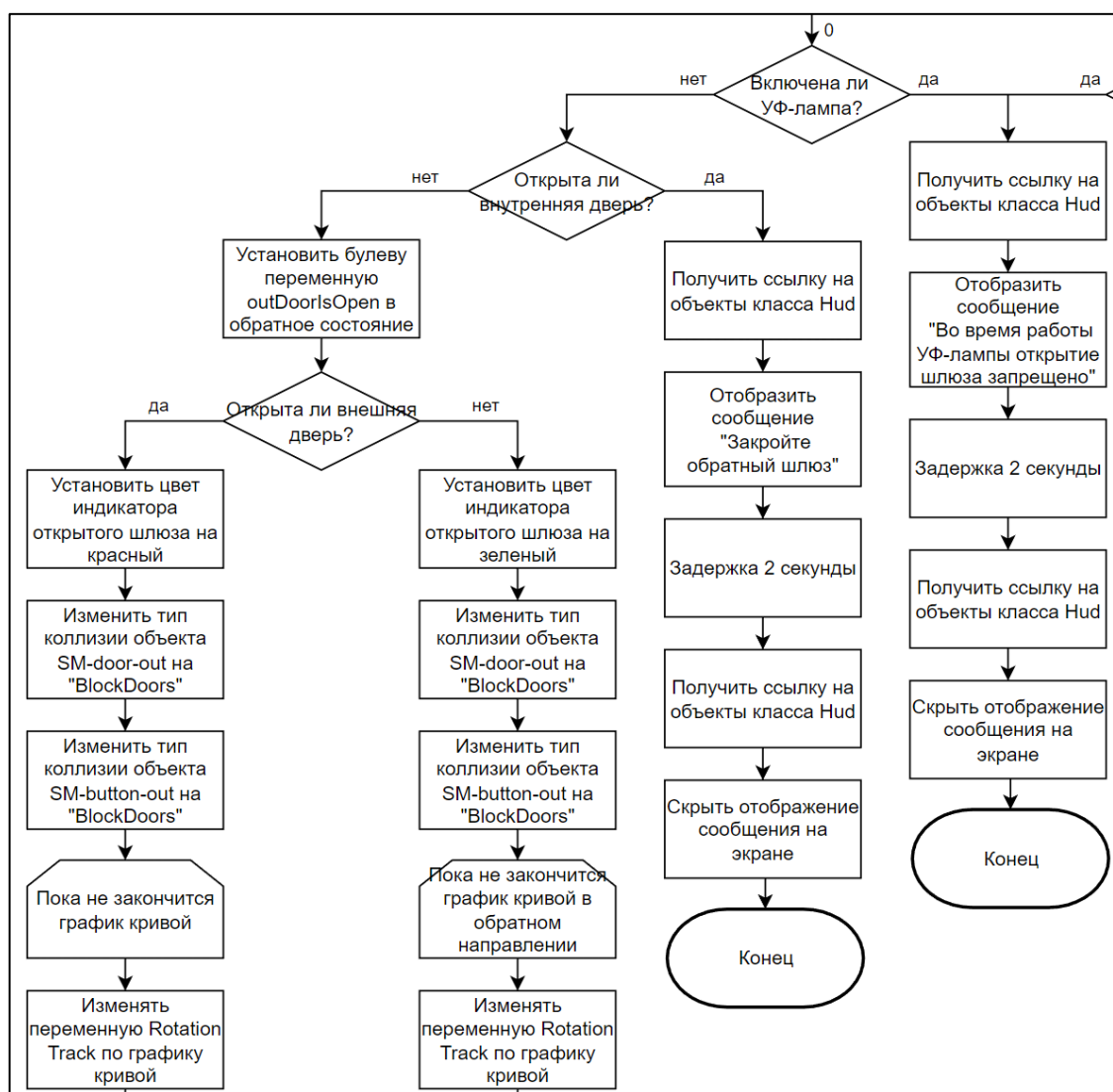


Рисунок 1.14 – Скриншот второй части блок-схемы скрипта шлюза передачи вещей

Если дверь закрыта, начинается идентичный предыдущему процесс открытия двери шлюза: производятся все аналогичные действия за исключением разницы в цвете светового индикатора – индикатор начинает светиться зеленым цветом. А график функции изменения относительного угла вращения дверцы проходит в обратном направлении движения. Недостающие на нижестоящем рисунке операции можно увидеть на после идущем рисунке 1.15.

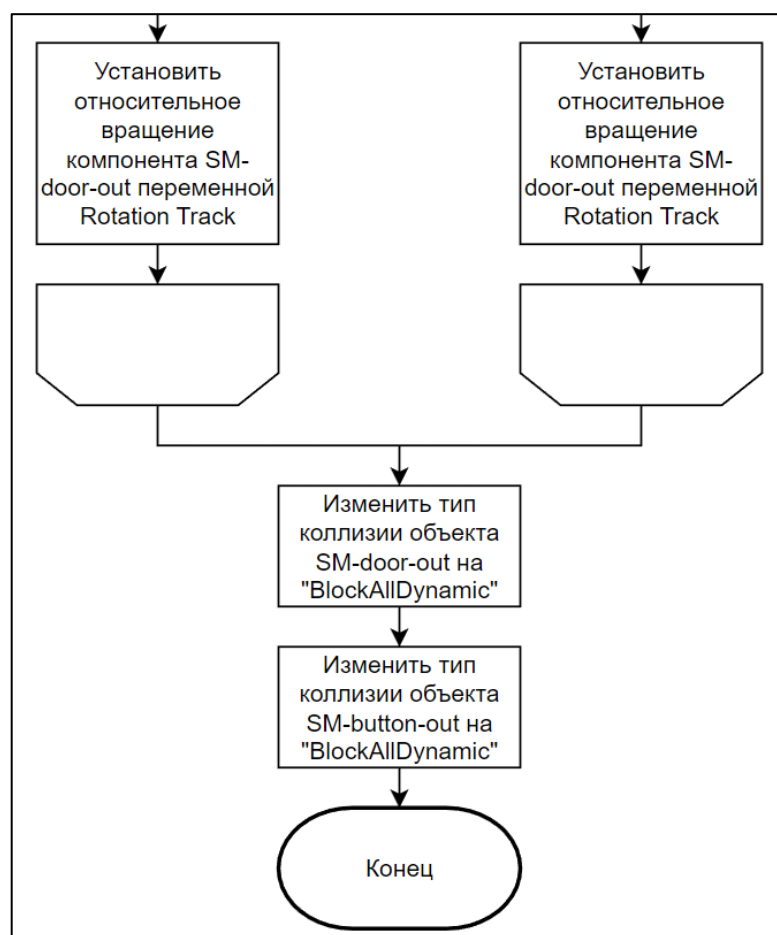


Рисунок 1.15 – Скриншот третьей части блок-схемы скрипта шлюза передачи вещей

На следующем рисунке 1.16 изображена первая половина графа операций, выполняемых в случае взаимодействия с внутренней дверью шлюза для передачи предметов в стерильное помещение.

Описание последовательности действий при взаимодействии с внутренней дверцей шлюза было принято не описывать подробно, поскольку принцип действия внутренней и внешней двери идентичен, разницу составляют лишь компоненты, которые участвуют в условных конструкциях и которые вращаются после прохождения всех условий, а именно: выключенной УФ-лампы и закрытой внешней дверцы шлюза передачи предметов, блокирующих возможность взаимодействия с внутренней дверцей шлюза.

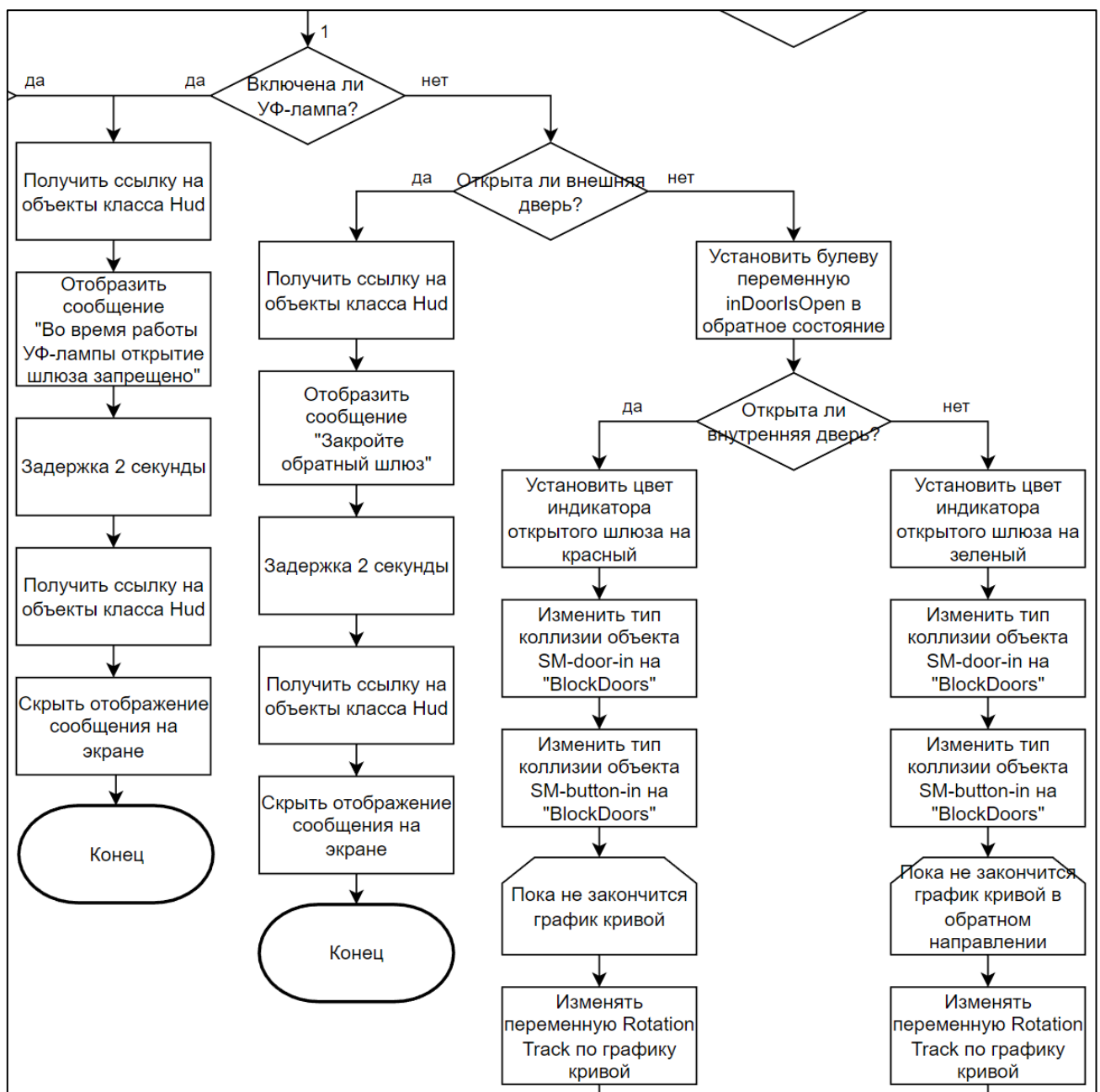


Рисунок 1.16 – Скриншот четвертой части блок-схемы скрипта шлюза передачи вещей

На рисунке 1.17 ниже изображена вторая половина графа операций, выполняемого при взаимодействии с внутренней дверцей шлюза передачи предметов.

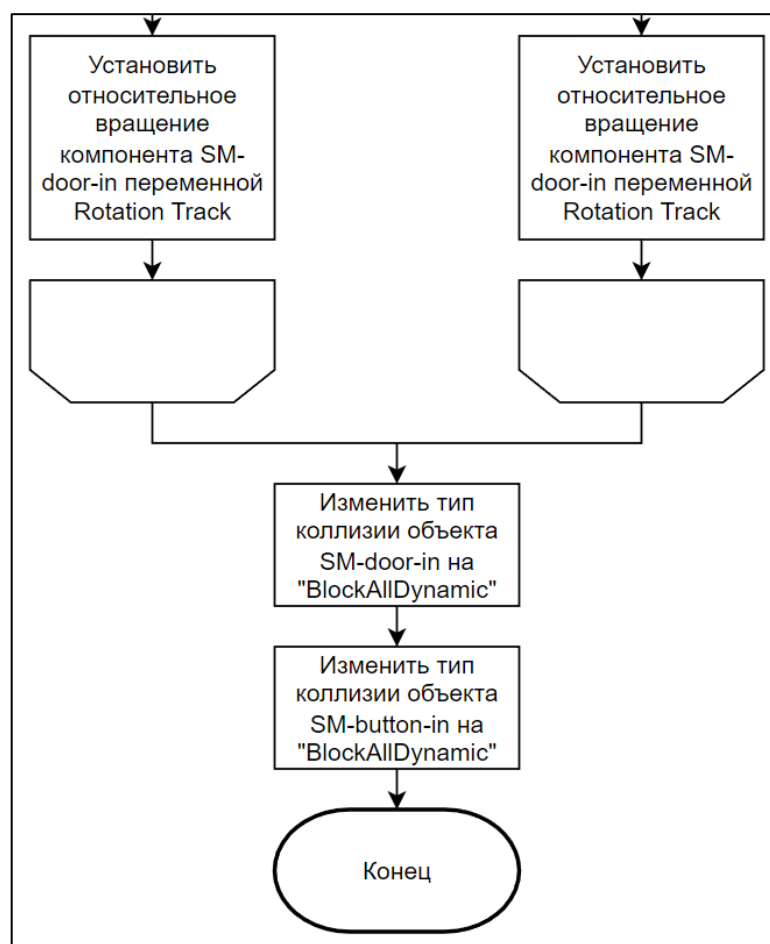


Рисунок 1.17 – Скриншот пятой части блок-схемы скрипта шлюза передачи вещей

При взаимодействии с внешней кнопкой включения ультрафиолетовой лампы для стерилизации шлюза передачи вещей выполняется последовательность операций, изображенной на рисунках 1.18-1.19.

В первую очередь проверяется, открыта ли любая из дверей. В случае того, что какая-то из дверей шлюза открыта, на экран выводится сообщение на две секунды, сообщающее, что включение УФ-лампы при открытом шлюзе запрещено. В ином случае полной герметизации у пользователя отключается возможность взаимодействия с дверцей и кнопкой используемой стороны шлюза с целью предотвращения неправильного поведения компонентов. После чего в цикле производится движение кнопки по установленной координате Y, имитирующее реалистичное поведение кнопки при нажатии на нее: сначала

кнопка вдавливается в приборную панель, затем возвращается на исходное положение. Данное поведение создано с помощью графика функции, изображенного на рисунке 2.5 (см. вторую главу).

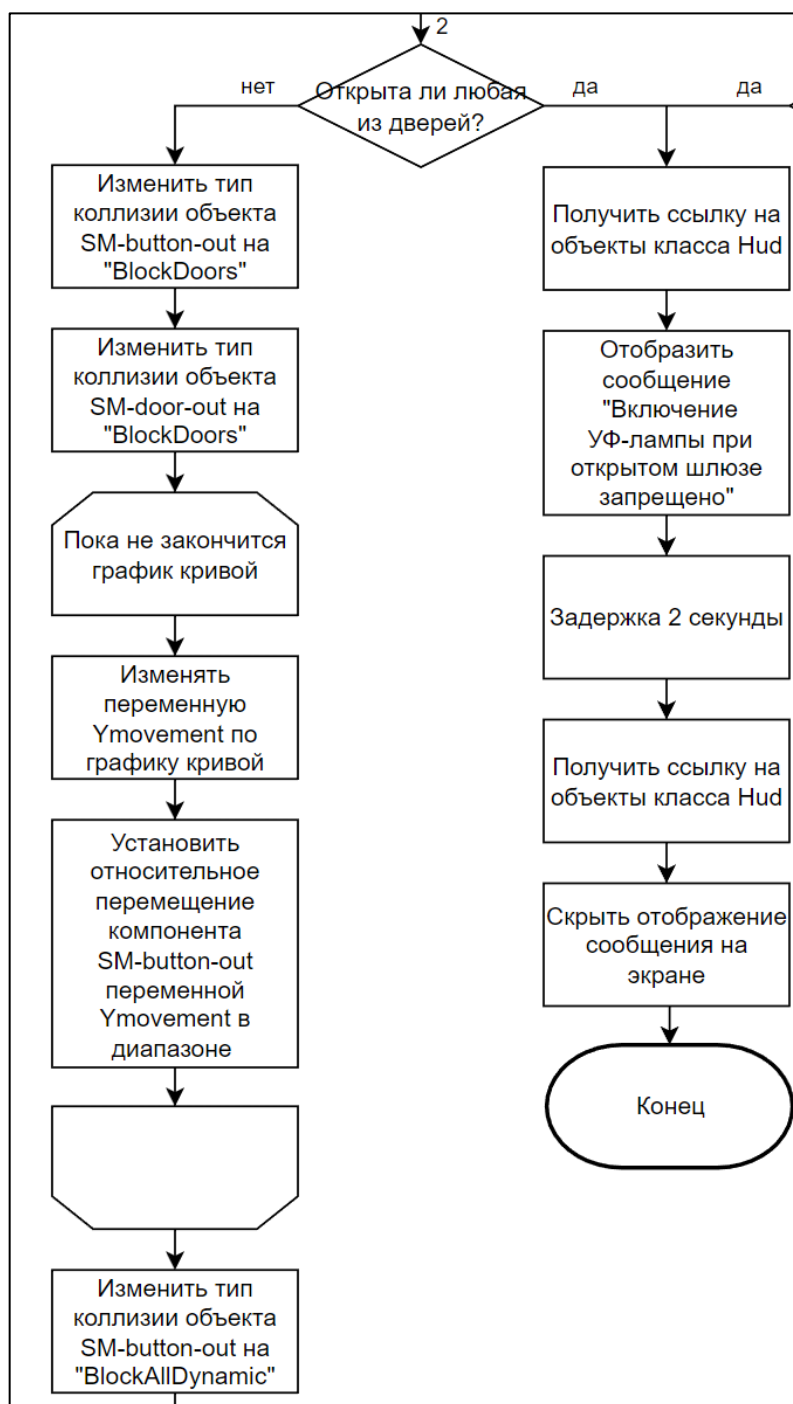


Рисунок 1.18 – Скриншот шестой части блок-схемы скрипта шлюза передачи вещей

После выполнения анимации нажатия кнопки специальная булева переменная, отвечающая за состояние работы ультрафиолетовой лампы меняет свое состояние на противоположное. После чего производится проверка, в зависимости от результатов которой лампа и индикатор работы лампы либо включаются и загораются фиолетовым соответственно, либо оба выключаются и завершают выполнение графа операций.

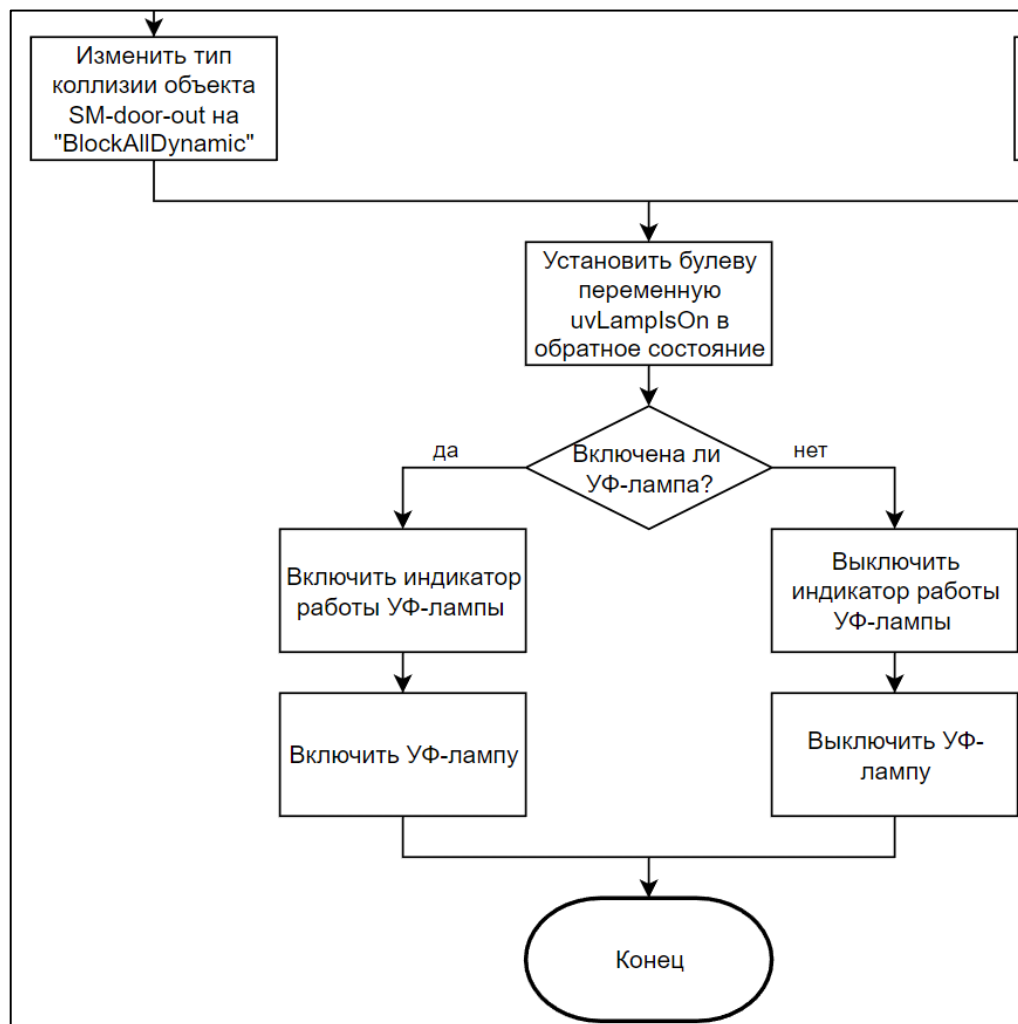


Рисунок 1.19 – Скриншот седьмой части блок-схемы скрипта шлюза передачи вещей

Наконец, последняя часть блок-схемы состоит из двух частей графа операций, изображенных на рисунках 1.20-1.21, отвечающих за поведение компонентов при взаимодействии с внутренней кнопкой.

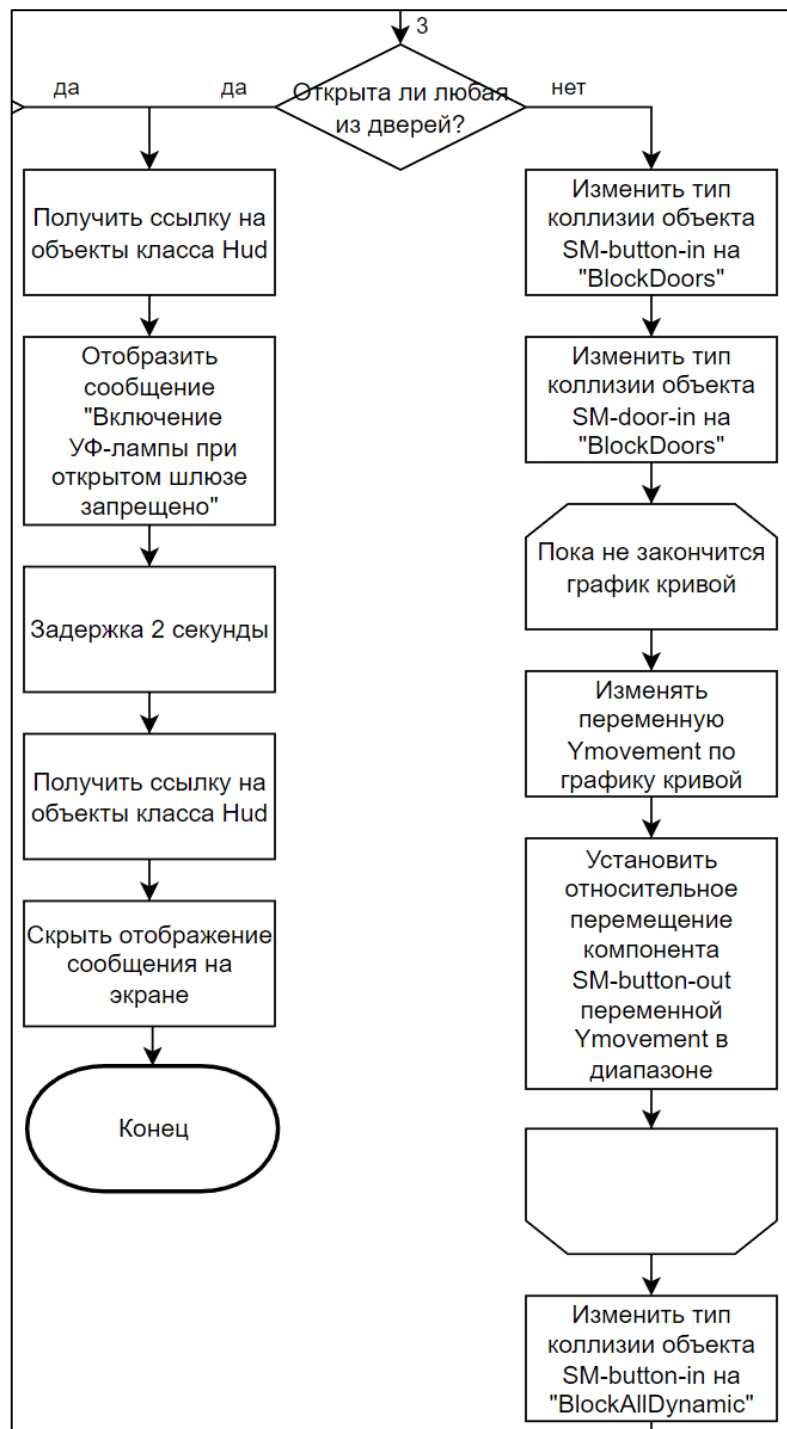


Рисунок 1.20 – Скриншот восьмой части блок-схемы скрипта шлюза передачи вещей

Последовательность действий идентична той, что производилась при взаимодействии с внешней кнопкой, различаются лишь компоненты со своими зеркальными аналогами между собой. Это означает, что вместо блокировки

взаимодействия с внешними компонентами происходит блокировка взаимодействия с внутренними компонентами.

На рисунке 1.21 изображена аналогичная последовательность действий, производимая после анимации нажатия внутренней кнопки, направленных на включение или выключение индикатора работы УФ-лампы и самой лампы в зависимости от ее актуального состояния.

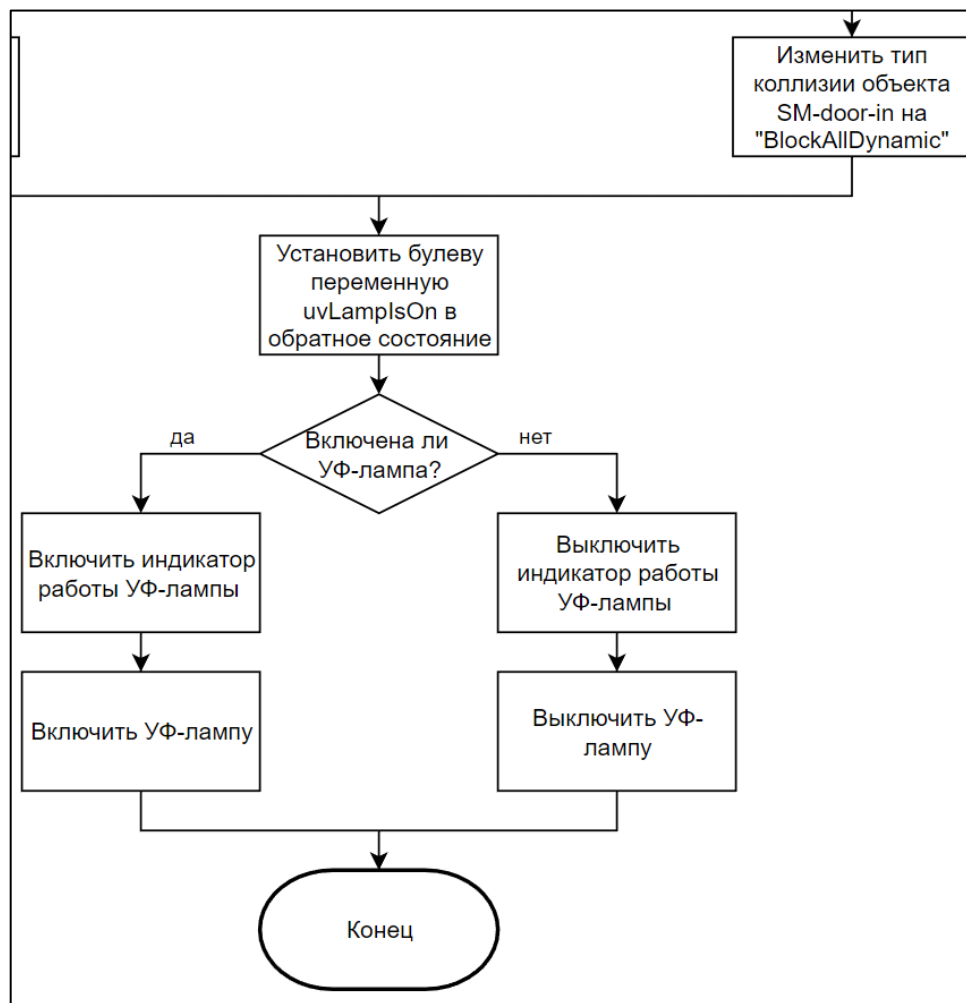


Рисунок 1.21 – Скриншот девятой части блок-схемы скрипта шлюза передачи вещей

2 Разработка логических сценариев виртуальных элементов симуляции

По причине подробного и исчерпывающего описания алгоритмов работы скриптов было принято решение не дублировать информацию в данной главе, а внести дополнения, связанные с технической и визуальной составляющей работы блюпринтов, подразумевающие собой графики функций различных переменных, а также изображения, демонстрирующие процесс работы элементов окружения в пространстве виртуальной химической лаборатории.

На рисунке 2.1 изображен график функции ArrowRotation, представляющий собой колебание стрелки манометра по установленной амплитуде, равной трем градусам. График построен по такому принципу, что движение стрелки зациклено с повторением до тех пор, пока не будет вызвано событие остановки этого таймлайна.

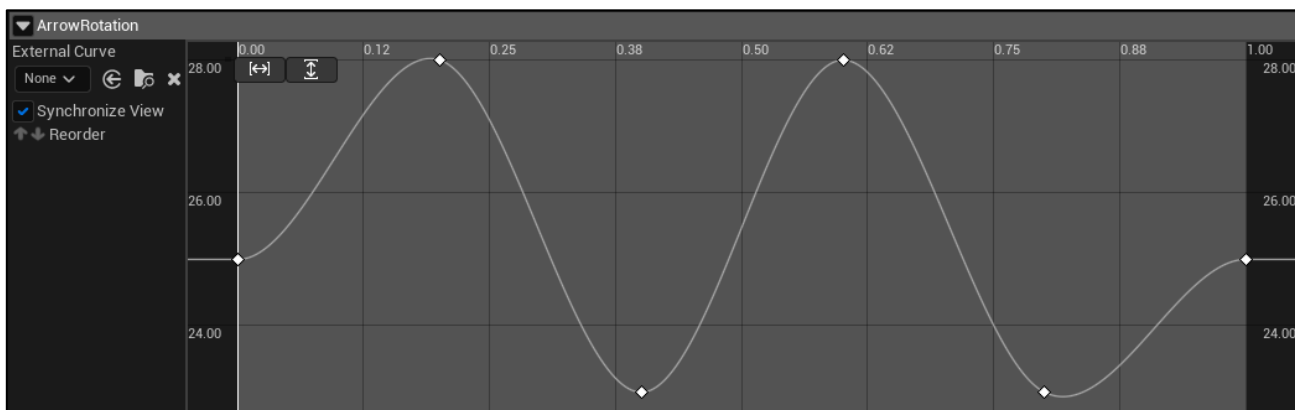


Рисунок 2.1 – Скриншот графика функции ArrowRotation

На следующем рисунке 2.2 представлен график функции ArrowStoppingRotation, благодаря которому по закрытии дверей и стабилизации давления в помещениях стрелка манометра плавно возвращается из положения колебаний около 13 паскаль в стабильное положение нуля паскаль. В градусах относительного вращения стрелки манометра этим значениям равны 25 и 45 градусов соответственно.

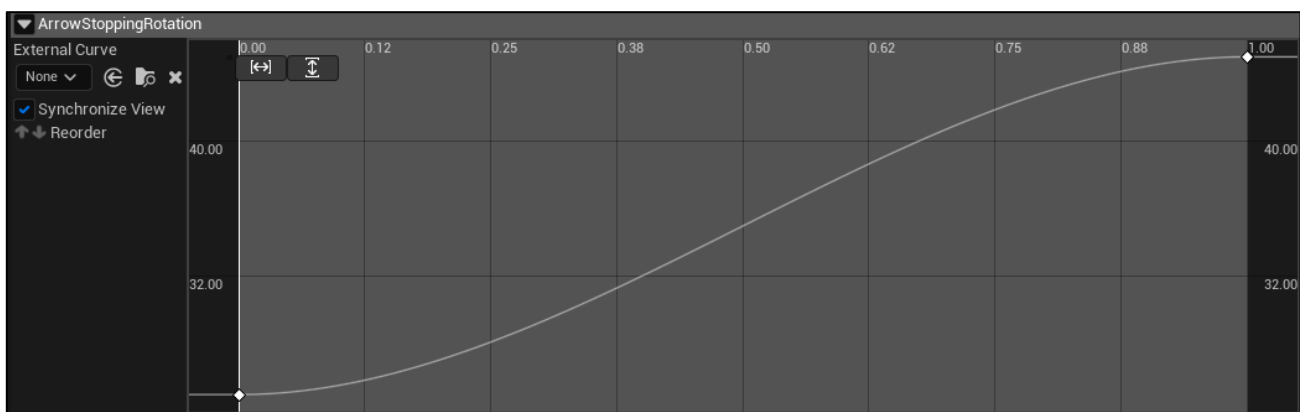


Рисунок 2.2 – Скриншот графика функции ArrowStoppingRotation

Далее изображен график функции DoorRotationZ, отвечающий за плавное изменение угла вращения двери от 0 до 130 градусов, изображенный на рисунке 2.3 ниже. Данный график используется во всех аналогичных герметичных дверях.

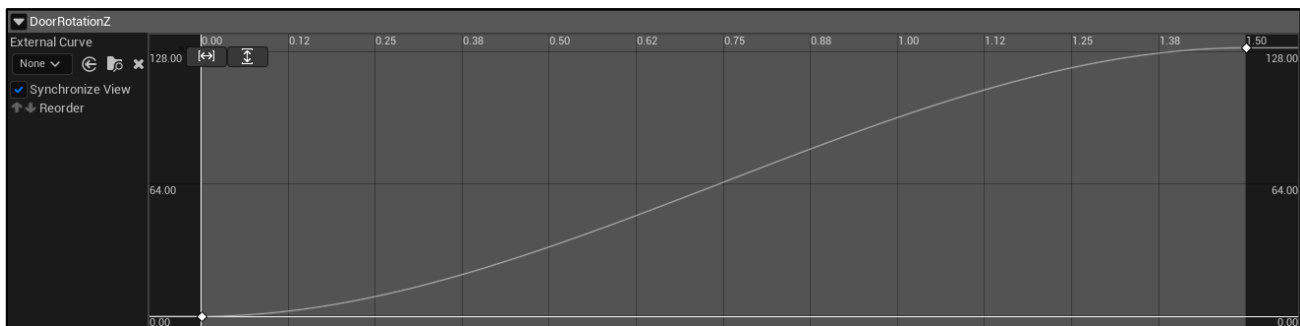


Рисунок 2.3 – Скриншот графика функции DoorRotationZ

Рисунок 2.4 изображает график функции RotationTrack, позволяющий правдоподобно и плавно открывать дверцы шлюза для передачи вещей, изменяя переменную относительного угла вращения с 0 до 75 градусов.

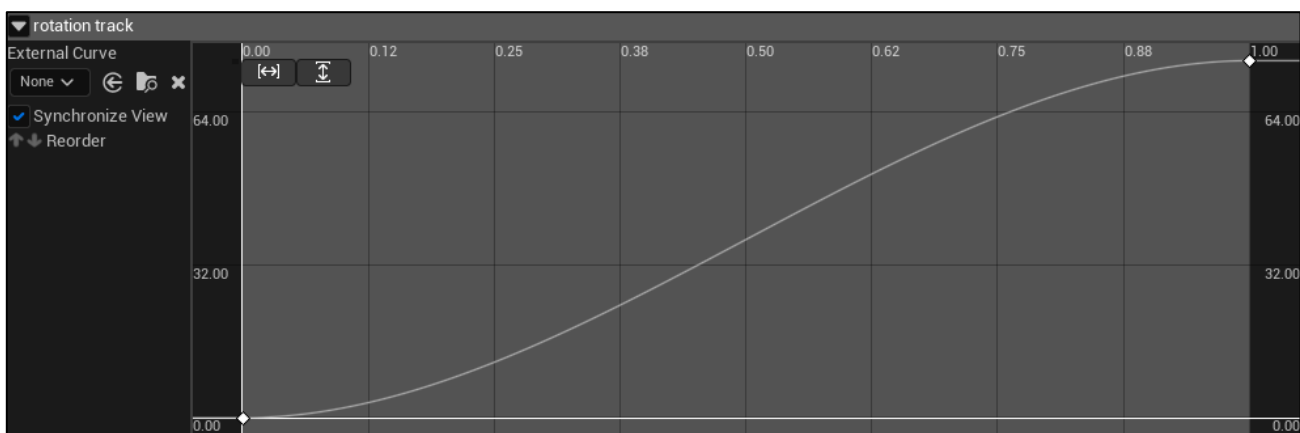


Рисунок 2.4 – Скриншот графика функции RotationTrack

Последний рисунок с графиком 2.5 демонстрирует график функции Ymovement, используемый для анимации правдоподобного нажатия кнопки на приборной панели шлюза для передачи предметов в стерильное помещение. Симметричный график начинает свое движение в нулевой координате и останавливается в точке со значением 1.

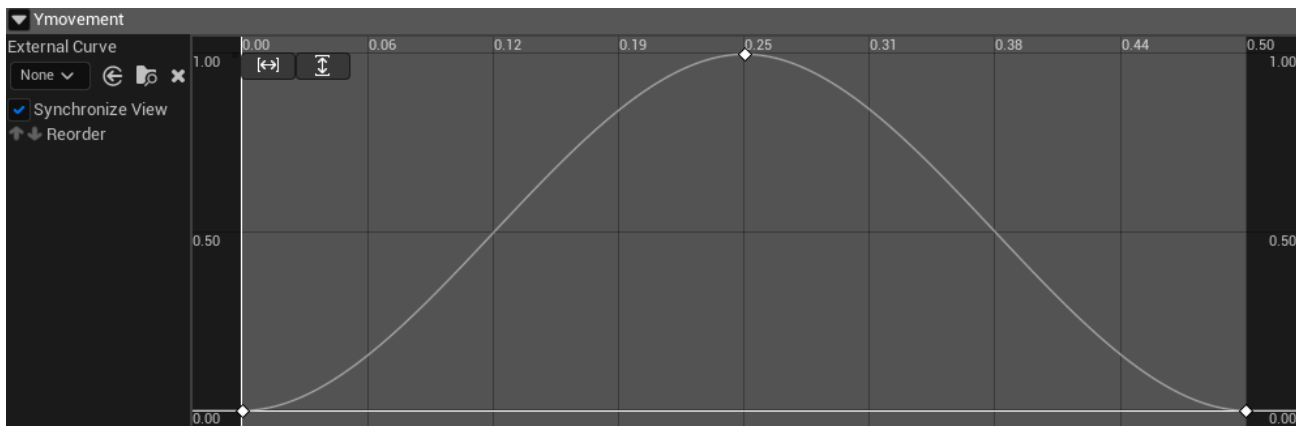


Рисунок 2.5 – Скриншот графика функции Ymovement

Далее на рисунке 2.6 изображена связь узлов таймлайна графика функции Ymovement с математической функцией Lerp, представляющей собой плавное изменение значения координаты Y компонента шлюза для передачи предметов в установленном диапазоне в зависимости от изменения альфы, являющейся переменной графика функции Ymovement в каждый момент времени выполнения анимации [7].

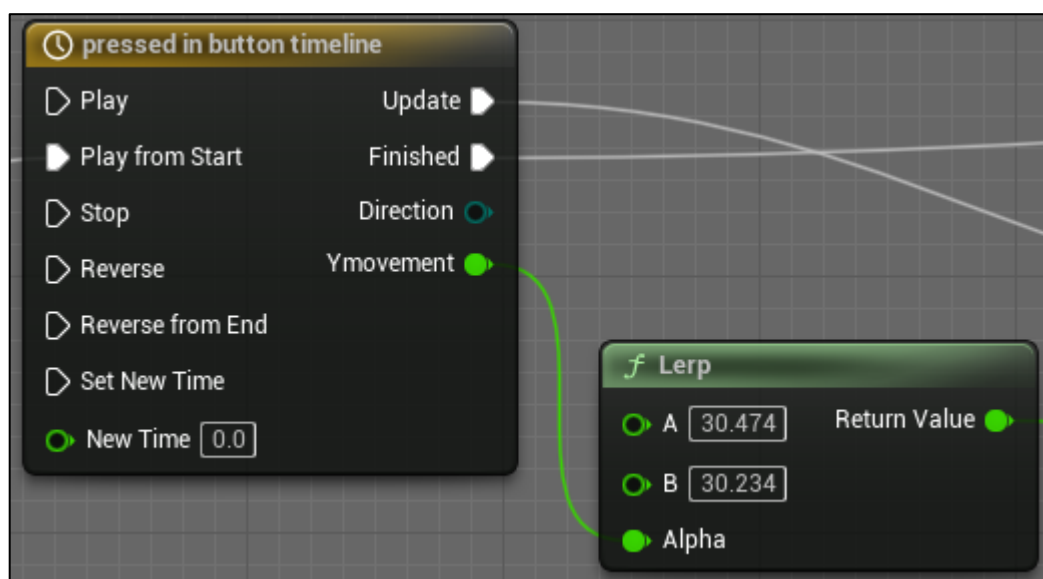


Рисунок 2.6 – Скриншот использования переменной графика функции Ymovement с математической функцией Lerp

ЗАКЛЮЧЕНИЕ

В результате выполнения данной научно-исследовательской работы были достигнуты поставленные цели и задачи, также были успешно построены и разобраны блок-схемы логических сценариев виртуальных элементов симуляции на основе связанных между собой графов операций и логических конструкций, которые представляют собой основную логику виртуальной лаборатории.

Методом сравнительного анализа были определены перспективные решения и реализованы требуемые логические сценарии трехмерных клонов, включая визуальное отображение скриптов, симулируемые процессы и варианты поведения пользователя в виртуальной среде. Были смоделированы удобные для восприятия графические схемы логических сценариев, наглядно демонстрирующие всевозможные варианты развития событий при взаимодействии пользователя с элементами окружения в разрабатываемой симуляции.

Для более наглядного восприятия все рассмотренные в рамках данной научной работы блок-схемы были также детально и последовательно описаны в письменной форме.

Были успешно разработаны сами скрипты поведения трехмерных клонов, которые будут использоваться в создаваемой фотореалистичной симуляции стерильной комнаты.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Unreal Engine – Официальный сайт [Электронный ресурс]. URL: <https://www.unrealengine.com/en-US> (дата обращения: 12.04.23).
2. Introduction to Blueprints Visual Scripting in Unreal Engine [Электронный ресурс]. URL: <https://docs.unrealengine.com/5.1/en-US/introduction-to-blueprints-visual-scripting-in-unreal-engine/> (дата обращения: 12.04.23).
3. Events in Unreal Engine [Электронный ресурс]. URL: <https://docs.unrealengine.com/5.1/en-US/events-in-unreal-engine/> (дата обращения: 12.04.23).
4. Components in Unreal Engine [Электронный ресурс]. URL: <https://docs.unrealengine.com/5.1/en-US/components-in-unreal-engine/> (дата обращения: 12.04.23).
5. Delay – Unreal Engine Documentation [Электронный ресурс]. URL: <https://docs.unrealengine.com/4.26/en-US/BlueprintAPI/Utilities/FlowControl/Delay/> (дата обращения: 12.04.23).
6. Timelines – Unreal Engine Documentation [Электронный ресурс]. URL: <https://docs.unrealengine.com/4.27/enUS/ProgrammingAndScripting/Blueprints/UserGuide/Timelines/> (дата обращения: 12.04.23).
7. Lerp – Unreal Engine Documentation [Электронный ресурс]. URL: <https://docs.unrealengine.com/4.27/en-US/BlueprintAPI/Math/Float/Lerp/> (дата обращения: 12.04.23).