



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных
технологий

Отчет по практической работе №1

по дисциплине «Тестирование и верификация ПО»

Выполнил:

Студент группы ИКБО-32-23

Мелива С.В.

Проверил:

Преподаватель

Ильичев Г.П.

СОДЕРЖАНИЕ

РТУ МИРЭА.....	1
СОДЕРЖАНИЕ	2
Техническое задание на разработку программы автоматизированного сбора системной информации с удаленного компьютера	4
1. Общие сведения	4
2. Цели и назначение создания автоматизированной системы	4
3. Характеристика объектов автоматизации.....	5
4. Требования к автоматизированной системе	5
4.1 Функциональные требования	5
4.2 Требования к производительности	5
4.3 Требования к интерфейсу	6
4.4 Требования по безопасности.....	6
5. Критерии приемки.....	6
5.1 Критерии приемки функционала:	6
5.2 Критерии приемки сетевого взаимодействия:	8
5.3 Критерии приемки многопоточности:	8
5.4 Критерии приемки сбора системной информации:	8
5.5 Критерии приемки пользовательского интерфейса:	9
5.6 Критерии приемки совместимости:	9
5.7 Тестирование сценариев использования:	9
5.8 Критерии приемки производительности:	10
6. Требования к документированию	10
7. Порядок контроля и приемки автоматизированной системы.....	10

8. Состав и содержание работ по созданию автоматизированной системы	12
9. Порядок разработки автоматизированной системы.....	12
10. Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу автоматизированной системы в действие	13
Документация на программный продукт “ LocalRetriever”	14
1. Описание программы.....	14
2. Инструкция пользователя.....	15
Техническое задание на программу рекреационного характера	23
1. Описание сути продукта и обоснование его разработки	23
2. Документация.....	23
Найденные ошибки с описанием того, как они были обнаружены.....	26
Проблема отсутствия критериев приемки	26
1. Спонтанное завершение игры	26
2. Невозможность переключения рубильников	27
Заключение	29

ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА РАЗРАБОТКУ ПРОГРАММЫ АВТОМАТИЗИРОВАННОГО СБОРА СИСТЕМНОЙ ИНФОРМАЦИИ С УДАЛЕННОГО КОМПЬЮТЕРА

1. Общие сведения

- **Наименование проекта:** "LocalRetriever".
- **Разработчик:** Индивидуальный разработчик.
- **Заказчик:** Пользователи ПК на базе Windows.
- **Исполнитель:** Программист на Python.
- **Основание для разработки:** Курсовая работа.

2. Цели и назначение создания автоматизированной системы

Целью создания системы автоматического сбора системной информации является разработка утилиты для персональных компьютеров, предназначенной для мониторинга удаленных устройств. Подсистема должна обеспечить простоту и удобство использования аудитории, нуждающейся в решениях по обслуживанию и поддержке множественных удаленных устройств локальной сети.

3. Характеристика объектов автоматизации

Объектом автоматизации данной подсистемы является сбор системных данных с удаленных компьютеров локальной сети, нуждающихся в обслуживании и мониторинге работоспособности.

4. Требования к автоматизированной системе

4.1 Функциональные требования

- Функция автоматизированного сбора системной информации:
 - Данные об аппаратном обеспечении.
 - Данные о программном обеспечении.
- Функция автоматизированного налаживания связи с удаленным компьютером.
- Функция автоматизированной передачи данных с удаленного компьютера.
- Наличие пользовательского интерфейса, визуализирующего процесс работы с ПО.

4.2 Требования к производительности

Программа должна обеспечивать сбор и передачу данных на ПК с минимальными системными требованиями (одноядерный процессор, 2 ГБ оперативной памяти).

4.3 Требования к интерфейсу

- GUI должен содержать окно с функцией выбора необходимой задачи: передать системную информацию, получить системную информацию.
- GUI должен содержать поля для ввода данных, необходимых для налаживания связи с удаленным компьютером: IP-адрес, порт.
- GUI должен содержать поле для текста, которое будет заполняться полученной с удаленного компьютера системной информацией.
- Графический пользовательский интерфейс должен быть интуитивно понятным, функциональным.

4.4 Требования по безопасности

- Программа должна работать в автономном режиме без взаимодействия с данными за пределами локальной сети.

5. Критерии приемки

5.1 Критерии приемки функционала:

- Главное окно:
 - При запуске приложения должно отображаться главное окно с заголовком «Приложение-сборщик»
 - Размер главного окна должен быть фиксирован и составлять 400 на 100 пикселей
 - В главном окне должны присутствовать две кнопки, открывающие окна «Отправитель» и «Получатель»

- Каждое окно должно открываться только в одном экземпляре
- Окно отправителя:
 - Окно должно иметь заголовок «Отправитель» и иметь фиксированный размер 350 на 200 пикселей
 - Окно должно содержать поля для ввода IP-адреса и порта, поля должны иметь значения по-умолчанию
 - Должна присутствовать кнопка «Отправить», инициирующая процесс сбора и отправки системной информации
 - Должен производиться сбор информации и системе (процессор, видеокарта, операционная система)
 - Собранные данные должны отправляться на указанный адрес и порт
 - Окно должно отображать соответствующие сообщения в случае успешной и неуспешной отправки данных
- Окно получателя:
 - Окно должно иметь заголовок «Получатель» и иметь фиксированный размер 350 на 400 пикселей
 - Окно должно содержать поля для ввода IP-адреса и порта, поля должны иметь значения по-умолчанию
 - Должна присутствовать кнопка «Получить данные», запускающая прослушивание указанного порта
 - В надлежащем поле должна корректно приниматься и отображаться полученная системная информация в формате JSON
 - Окно должно отображать соответствующее сообщение в случае неудавшегося получения информации

5.2 Критерии приемки сетевого взаимодействия:

- Отправка и прием данных должны производиться через TCP-сокеты в пределах локальной сети
- Данные должны передаваться в формате JSON с кодировкой UTF-8
- Должна поддерживаться работа на localhost (127.0.0.1)
- Должна присутствовать обработка ошибок соединения
- Должно производиться корректное закрытие сокетов после использования

5.3 Критерии приемки многопоточности:

- Сетевые операции должны выполняться в отдельных потоках
- GUI не должен блокироваться во время сетевых операций
- Кнопки должны временно отключаться на время выполнения операций
- Должна присутствовать корректная обработка завершения потоков

5.4 Критерии приемки сбора системной информации:

- Должны корректно определяться характеристики процессора: имя, архитектура, кол-во ядер
- Должны корректно определяться характеристики видеокарты: имя, разрешение

- Должна корректно определяться информация об операционной системе: название, версия

5.5 Критерии приемки пользовательского интерфейса:

- Все окна должны корректно отображаться без обрезания элементов GUI и текста
- Текстовые поля должны поддерживать ввод и отображение кириллицы
- Кнопки должны иметь простые и логичные названия
- Выводимые сообщения об ошибках должны доступно описывать проблему

5.6 Критерии приемки совместимости:

- Программный продукт должен работать на ОС Windows
- ПП должен работать с разными версиями Python 3.x

5.7 Тестирование сценариев использования:

- Отправка и прием данных на одном компьютере (localhost)
- Отправка данных с некорректными параметрами (неверный IP-адрес / порт)
- Многократное открытие / закрытие окон
- Параллельная работа отправителя и получателя

5.8 Критерии приемки производительности:

- Отклик интерфейса не должен задерживаться при сетевых операциях
- Потребление памяти не должно превышать 100 МБ
- Должны отсутствовать утечки ресурсов

6. Требования к документированию

Должны быть подготовлены следующие документы:

Руководство пользователя: Подробное описание функционала программы, инструкции по установке, настройке и использованию.

Техническая документация: Описание алгоритмов работы программы, требования к окружению и инструкции по развертыванию.

7. Порядок контроля и приемки автоматизированной системы

Контроль осуществляется в процессе тестирования программы.

Для полного тестирования должно быть подготовлено тестовое окружение:

- Минимум 2 ПК в пределах локальной сети с ОС Windows
- Сетевое соединение между ПК
- Установленный на оба ПК Python версии 3.x
- Установленные на оба ПК зависимости: PyQt5, wmi, pywin32

Для проверки соответствия программного решения заявленным требованиям также необходимы: техническое задание и руководство пользователя.

На этапе тестирования производятся следующие операции:

- Проверка сетевого соединения на обоих ПК
- Установка зависимостей
- Запуск приложения на обоих ПК
 - Проверка заголовка и размеров окна
 - Проверка наличия и функционирования кнопок «Открыть окно отправителя» и «Открыть окно получателя»
 - Проверка заголовков, полей и размеров окон «Отправителя» и «Получателя»
- Многократное открытие и закрытие окон приложения
- Параллельное взаимодействие с окнами «Получатель» и «Отправитель»
- Проверка успешной передачи системной информации между ПК в локальной сети
 - Ввод соответствующих IP-адресов и портов на обоих ПК
 - Пересылка и проверка полученных данных
- Проверка успешного считывания данных об устройстве на одном ПК (localhost)
 - Проверка параллельного сбора и вывода системной информации на устройстве с использованием IP-адреса и порта, заполненных по-умолчанию
 - Проверка корректности данных
- После успешного прохождения всех тестов программа считается готовой к приёму заказчиком.

8. Состав и содержание работ по созданию автоматизированной системы

- Анализ существующих решений.
- Разработка концепции утилиты для автоматической передачи системных данных с удаленных компьютеров.
 - Реализация базовой функциональности (налаживание связи между устройствами, сбор и передача информации).
 - Создание пользовательского интерфейса.
 - Тестирование производительности и стабильности системы.
 - Оптимизация производительности программы и исправление ошибок.
- Подготовка финальной версии программы (в формате исполняемого файла).

9. Порядок разработки автоматизированной системы

1. **Анализ требований:** Сбор информации и проектирование функционала ПО.
2. **Прототипирование:** Разработка базовой функциональности.
3. **Основная разработка:** Добавление интерфейса.
4. **Тестирование и оптимизация:** Тестирование на производительность и исправление ошибок.

10. Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу автоматизированной системы в действие

До ввода программы в эксплуатацию необходимо:

Провести полное тестирование программы на целевой платформе (Windows).

Настроить среду запуска программы.

Подготовить техническую документацию и руководство пользователя

ДОКУМЕНТАЦИЯ НА ПРОГРАММНЫЙ ПРОДУКТ “ LOCALRETRIEVER”

1. Описание программы

LocalRetriever — это программное обеспечение для ПК под управлением Windows, предназначенное для сбора системной информации с устройств, находящихся в одной локальной сети. Цель программы — упрощение процесса мониторинга локальных компьютерных систем. Приложение имеет простой и интуитивно понятный интерфейс.

Основные элементы:

- Главное меню – окно под названием «Приложение-сборщик», в данном окне представлены две кнопки: «Открыть окно отправителя» и «Открыть окно получателя».
- Окно «Отправитель» - представляет пользователю два подписанных поля для заполнения: «Введите IP-адрес удаленного компьютера: », предназначенное для указания IP-адреса компьютера, запрашивающего системную информацию данного устройства, и «Введите порт: », предназначенное для указания порта.
- Окно «Получатель» - представляет пользователю два подписанных поля для заполнения: «Введите IP-адрес вашего компьютера: », предназначенное для указания IP-адреса компьютера пользователя, запрашивающего информацию от удаленного устройства, и «Введите порт: », предназначенное для указания порта; также данное окно содержит не редактируемое поле для текста под названием «Информация о системе: », после получения и обработки

запрашиваемых данных это поле заполняется текстовым отчетом о конфигурации удаленной системы.

2. Инструкция пользователя

Разработанная программа автоматизированного сбора системной информации с удаленного компьютера может быть запущена через приложение командной строки путем ввода следующих команд:

1. Необходимо ввести команду перехода в директорию, где находятся все три файла программы («main.py», «sender.py», «receiver.py»). Команда выглядит следующим образом: «*cd путь до директории*».
2. Далее необходимо произвести запуск программы путем ввода команды: «python main.py» (Важно отметить, что для работы данной команды необходимо наличие на ПК интерпретатора Python)

Альтернативно разработанное ПО можно привести в работу через исполняемый файл с расширением «.exe». Для этого необходимо сперва произвести компиляцию программы при помощи команды, прописываемой в приложение командной строки:

1. Необходимо ввести команду перехода в директорию, где находятся все три файла программы («main.py», «sender.py», «receiver.py»). Команда выглядит следующим образом: «*cd путь до директории*».
2. Далее необходимо выполнить компиляцию при помощи команды: «pyinstaller --onefile --add-data "sender.py; receiver.py" main.py» (Для работы данной команды необходимо наличие на ПК: интерпретатора Python, пакета PyInstaller)

После запуска приложения на экран монитора пользователя выводится окно главного модуля программы под названием «*Приложение-сборщик*» (окно показано на рисунке 1). В данном окне пользователю необходимо

выбрать одну из опций: «Открыть окно отправителя», «Открыть окно получателя»; первая опция приводит к запуску модуля отправителя, который предоставляет возможность произвести сбор системной информации данного устройства и передать ее удаленному компьютеру, вторая опция приводит к запуску модуля получателя, который предоставляет возможность принять входящее соединение от удаленного компьютера, получить его системные данные и вывести их на экран монитора.

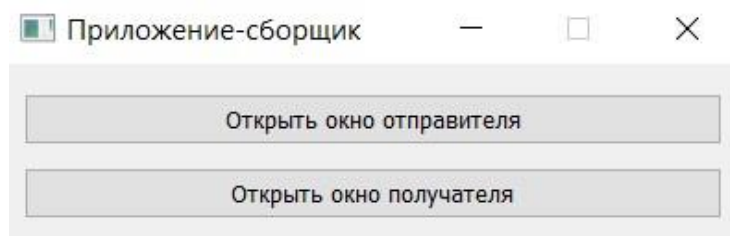


Рисунок 1 – Окно главного модуля программы, «Приложение-сборщик»

- При необходимости переслать системные данные: В окне модуля отправителя (окно показано на рисунке 2) от пользователя требуется заполнение двух полей: в поле под строкой «Введите IP-адрес удаленного компьютера: » вписывается IP-адрес устройства-получателя (удаленного устройства), в поле под строкой «Введите порт: » вписывается порт, который будет задействован. После заполнения вышеупомянутых полей и нажатия кнопки «Отправить» программа произведет попытку сбора и передачи системной информации удаленному компьютеру.

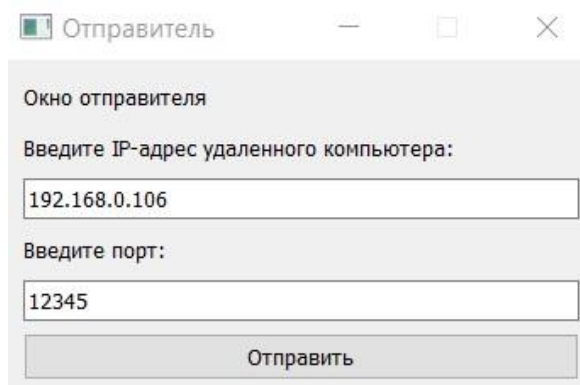
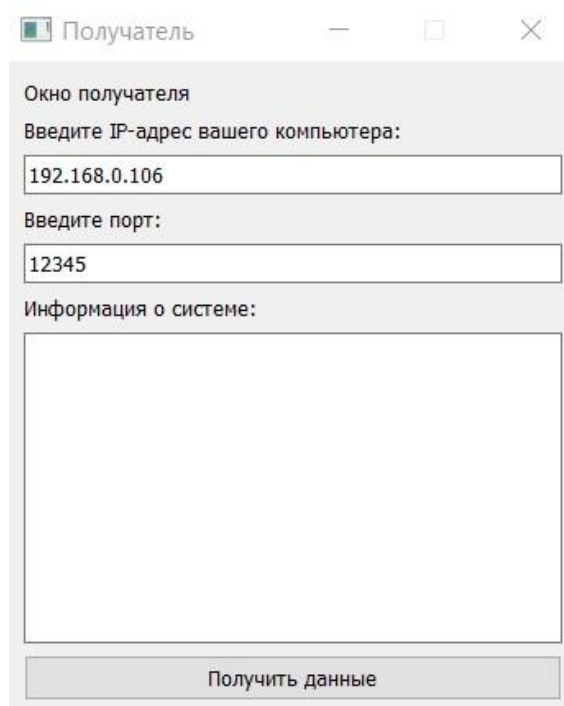


Рисунок 2 – Окно модуля отправителя с примером корректного заполнения полей

- При необходимости получить системные данные: В окне модуля получателя (окно показано на рисунке 3) от пользователя требуется заполнение двух полей: в поле под строкой *«Введите IP-адрес вашего компьютера:»* вписывается IP-адрес устройства-получателя (данного устройства), в поле под строкой *«Введите порт:»* вписывается порт, который будет задействован. После заполнения вышеупомянутых полей и нажатия кнопки *«Получить данные»* программа перейдет в режим ожидания входящего соединения; после принятия сигнала, налаживания связи и получения данных, в не редактируемом поле под строкой *«Информация о системе:»* появится текст, содержащий системную информацию удаленного компьютера (как показано на рисунке 4).



The image shows a Windows-style window titled "Получатель" (Receiver). Inside the window, there is a section titled "Окно получателя" (Receiver window). Below this title, there are two input fields. The first field is labeled "Введите IP-адрес вашего компьютера:" (Enter your computer's IP address:) and contains the text "192.168.0.106". The second field is labeled "Введите порт:" (Enter port:) and contains the text "12345". Below these fields, there is a label "Информация о системе:" (System information:) followed by a large, empty rectangular box. At the bottom of the window, there is a button labeled "Получить данные" (Get data).

Рисунок 3 – Окно модуля получателя с примером корректного заполнения полей

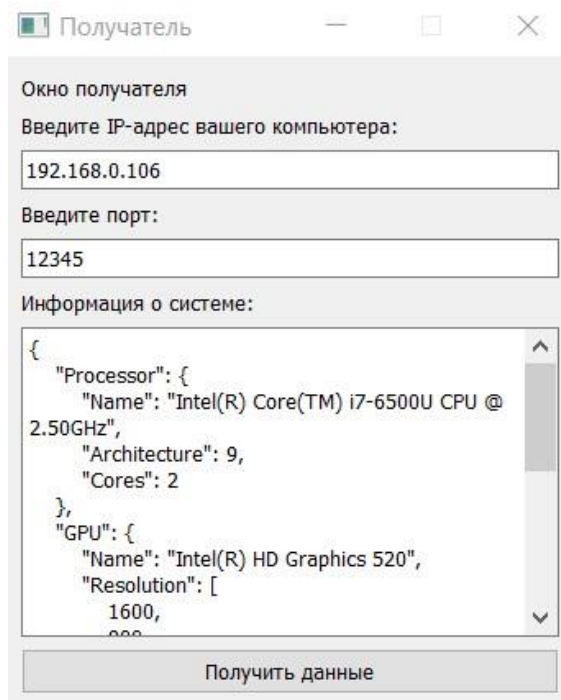


Рисунок 4 – Окно модуля получателя с системной информацией удаленного компьютера

- При необходимости получить системные данные текущего устройства: От пользователя требуется использовать данные, предлагаемые программой по-умолчанию, *IP-адрес принимает значение: 127.0.0.1, Порт принимает значение: 12345.*

Перечень ошибок внесенных в программный продукт

Ошибка №1

Тип ошибки: Логическая ошибка.

Описание ошибки: В файле программы «sender.py» в методе «get_system_info» на 133-й строке кода содержится ошибка, записывающая в переменную «processor_architecture» количество ядер процессора вместо архитектуры, в следствие чего в поле «Информация о системе: » окна модуля получателя отображается некорректная системная информация (см. листинг 1).

Листинг 1

<p>Ошибка: processor_architecture = processor_info.NumberOfCores</p> <p>Должно быть: processor_architecture = processor_info.Architecture</p>

Ошибка №2

Тип ошибки: Логическая ошибка.

Описание ошибки: В файле программы «sender.py» в методе «get_system_info» на 135-й строке кода содержится ошибка, записывающая в переменную «processor_cores» архитектуру процессора вместо количества ядер, в следствие чего в поле «Информация о системе: » окна модуля получателя отображается некорректная системная информация (см. листинг 2).

Листинг 2

<p>Ошибка: processor_cores = processor_info.Architecture</p> <p>Должно быть: processor_cores = processor_info.NumberOfCores</p>

Ошибка №3

Тип ошибки: Логическая ошибка.

Описание ошибки: В файле программы «main.py» в методе «__init__» класса «MainWindow» на 33-й строке кода содержится ошибка, вызывающая окно «Получатель» при нажатии кнопки «Открыть окно отправителя» (см. листинг 3).

Листинг 3

Ошибка:

```
send_button.clicked.connect(self.open_receiver_window)
```

Должно быть:

```
send_button.clicked.connect(self.open_sender_window)
```

Ошибка №4

Тип ошибки: Логическая ошибка.

Описание ошибки: В файле программы «main.py» в методе «__init__» класса «MainWindow» на 39-й строке кода содержится ошибка, вызывающая окно «Отправитель» при нажатии кнопки «Открыть окно получателя» (см. листинг 4).

Листинг 4

Ошибка:

```
receive_button.clicked.connect(self.open_sender_window)
```

Должно быть:

```
receive_button.clicked.connect(open_receiver_window)
```

Ошибка №5

Тип ошибки: Интерфейсная ошибка.

Описание ошибки: В файле программы «sender.py» в методе «get_system_info» на 154-й строке кода содержится ошибка, отображающая некорректный текст в поле «Информация о системе:» (см. листинг 5).

Листинг 5

```
Ошибка:
"OS": {
    "Name": gpu_name,
    "Resolution": gpu_resolution
}

Должно быть:
"GPU": {
    "Name": gpu_name,
    "Resolution": gpu_resolution
}
```

Ошибка №6

Тип ошибки: Интерфейсная ошибка.

Описание ошибки: В файле программы «sender.py» в методе «get_system_info» на 158-й строке кода содержится ошибка, отображающая некорректный текст в поле «Информация о системе:» (см. листинг 6).

Листинг 6

```
Ошибка:
"GPU": {
    "Name": os_name,
    "Version": os_version
}

Должно быть:
"OS": {
    "Name": os_name,
    "Version": os_version
}
```

Ошибка №7

Тип ошибки: Сетевая ошибка.

Описание ошибки: В файле программы «sender.py» в методе «destination» на 107-й строке кода содержится ошибка, увеличивающая величину введенного пользователем порта на единицу, делая тем самым соединение с удаленным устройством невозможным для не располагающего информацией об ошибке пользователя (см. листинг 7).

Листинг 7

Ошибка:

```
receiver_port = int(self.port_input.text()) + 1
```

Должно быть:

```
receiver_port = int(self.port_input.text())
```

Ошибка №8

Тип ошибки: Сетевая ошибка.

Описание ошибки: В файле программы «receiver.py» в методе «process_data» на 125-й строке кода содержится ошибка, увеличивающая величину введенного пользователем порта на единицу, делая тем самым соединение с удаленным устройством невозможным для не располагающего информацией об ошибке пользователя (см. листинг 8).

Листинг 8

Ошибка:

```
receiver_port = int(self.port_input.text()) + 1
```

Должно быть:

```
receiver_port = int(self.port_input.text())
```

ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА ПРОГРАММУ РЕКРЕАЦИОННОГО ХАРАКТЕРА

1. Описание сути продукта и обоснование его разработки

Игрок управляет рукой, при помощи которой он должен включить 4 рубильника и восстановить питание в помещении. Щиток старый и поэтому провода в нём часто оголены, при включении рубильника включаются вольтметры, отражающие прогресс, и провода принадлежащие рубильнику. Если игрок касается рукой включенных проводов, то умирает и игра начинается с начала.

Разработка обоснована спросом на простые, но захватывающие игры с минималистичным дизайном и напряженным геймплеем. Концепция идеально подходит для РС-платформы, где точное управление мышью создает необходимое ощущение контроля и напряжения.

2. Документация

Стартовая катсцена

Кнопка пропустить

При зажатии Е, в правой нижней части вокруг значка Е начинает заполняться зелёная окружность, если она полностью заполняется происходит пропуск катсцены.

Элементы игры и их функционал

Кнопка старта

При нажатии на неё начинается игра. Свет вырубается и игрок начинает видеть только силуэт руки и небольшую сильно затемненную область вокруг неё.

Рука

Её передвижения привязаны к мышке игрока, она плавно перемещается к положению мыши используя эффект резинки.

Переключатели

При наведении Руки на переключатель появляется белый силуэт переключателя с направлением активации переключателя, и силуэт проводов подключенных к нему.

При зажатии ЛКМ на переключателе пока рука наведена на него, активируется анимация руки схватить.

Если начать движение мыши в сторону направления активации, то Рука начнёт дрожать.

Если продолжить движение мыши на расстояние {diastansToActivate}, то переключатель переводится в активное состояние, а рука отлетает в сторону положения мыши.

В включённом состоянии по проводам подсоединённым к переключателю начинает течь ток.

Провода

Если рука касается провода пока по ним течёт ток, то игрок проигрывает и сцена перезапускается.

Вольтметры

Пока по проводам подсоединенным к вольтметру течёт ток, экран вольтметра светиться.

Титры

Кнопка пропустить

При зажатии Е, в правой нижней части вокруг значка Е начинает заполняться зелёная окружность, если она полностью заполняется происходит пропуск катсцены.

НАЙДЕННЫЕ ОШИБКИ С ОПИСАНИЕМ ТОГО, КАК ОНИ БЫЛИ ОБНАРУЖЕНЫ

Проблема отсутствия критериев приемки

В рамках второй части практической работы требовалось провести тестирование программного продукта методом «черного ящика». Однако, в процессе выполнения работы была выявлена проблема: в техническом задании к продукту не были прописаны порядок контроля и приемки автоматизированной системы. Без формализованных требований к ПО корректное тестирование становится невозможным, а продукт не подлежит приему, так как:

- у продукта отсутствуют четкие критерии успешности выполнения функций;
- отсутствуют сценарии тестирования;
- становится невозможна объективная оценка работоспособности программы, а также ее соответствие ожидаемым требованиям.

Тем не менее, в меру возможности было проведено частичное тестирование продукта с целью выявления части ошибок.

1. Спонтанное завершение игры

Идентификатор.

ТС-001

Название.

Спонтанное завершение игры

Описание.

Во время игрового процесса и взаимодействия с рубильниками наступает «гейм-овер».

Предварительные условия:

- Приложение запущено
- Игра начата

Как обнаружить: провзаимодействовать с основными механиками игры.

Ожидаемый результат: поочередное переключение рубильников активирует ток в проводах; активация всех четырех рубильников приводит к «победе» и завершению игры.

Фактический результат: в процессе переключения рубильников один из них спонтанно активирует «гейм-овер».

Статус.

Failed

2. Невозможность переключения рубильников

Идентификатор.

ТС-002

Название.

Невозможность переключения рубильников

Описание.

При попытке переключения рубильников механика взаимодействия с ними через передвижение игроком мыши по экрану не работает корректно.

Предварительные условия:

- Приложение запущено
- Игра начата
- Мышь пользователя наведена на один из рубильников

Как обнаружить: произвести попытку переключения рубильника.

Ожидаемый результат: модель рубильника следует за движением мыши игрока и тем самым переходит из состояния «выкл» в состояние «вкл».

Фактический результат: модель рубильника не реагирует на движение мыши по направлению к состоянию «вкл», либо реагирует с задержкой, приводя тем самым к касанию проводов и «гейм-оверу».

Статус.

Failed

ЗАКЛЮЧЕНИЕ

В процессе работы над первой частью практической работы было успешно разработано техническое задание (ТЗ), а также сопутствующая документация программного продукта. Данная работы поспособствовала пониманию процесса формализации требований и подготовки продукта к тестированию. Был проведен анализ продукта с целью выявления ошибок. В процессе работы над второй частью практической работы была организована проверка программного продукта, созданного другой командой, с использованием метода «черного ящика».