ПРОГРАММА И МЕТОДИКА ИСПЫТАНИЙ ДЛЯ ПРОГРАММЫ «АВТОГОНКИ»

СОДЕРЖАНИЕ

[1 ОБЪЕКТ ИСПЫТАНИЙ 3](#_Toc135453733)

[1.1 Наименование программного обеспечения 3](#_Toc135453734)

[1.2 Область применения 3](#_Toc135453735)

[1.3 Обозначение испытуемого программного обеспечения 3](#_Toc135453736)

[2 ЦЕЛЬ ИСПЫТАНИЙ 4](#_Toc135453737)

[3 ТРЕБОВАНИЯ К ПРОГРАММЕ 5](#_Toc135453738)

[3.1 Требования к отдельным модулям программы 5](#_Toc135453739)

[3.1.1 Модуль генерации новых противников 5](#_Toc135453740)

[3.1.2 Модуль перемещения игрока 5](#_Toc135453741)

[3.1.3 Модуль оперирования счётом 5](#_Toc135453742)

[3.2 Функциональные требования 5](#_Toc135453743)

[3.3 Требования к производительности 6](#_Toc135453744)

[3.4 Требования к интерфейсу 7](#_Toc135453745)

[3.4.1 Требования к размещению пользовательского интерфейса 7](#_Toc135453746)

[3.4.2 Требования к наблюдаемости состояния системы 7](#_Toc135453747)

[3.4.3 Требования к гибкости и эффективности использования 7](#_Toc135453748)

[3.4.4 Требования к справочной информации 8](#_Toc135453749)

[3.4.5 Требования к системе распознавания, диагностики и устранения ошибок 8](#_Toc135453750)

[3.4.6 Требования к предотвращению ошибок 8](#_Toc135453751)

[4 ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ 9](#_Toc135453752)

[4.1 Состав программной документации, предъявляемой на испытания 9](#_Toc135453753)

[4.2 Специальные требования 9](#_Toc135453754)

[5 СРЕДСТВА И ПОРЯДОК ИСПЫТАНИЙ 10](#_Toc135453755)

[5.1 Средства испытаний 10](#_Toc135453756)

[5.1.1 Технические средства, используемые во время испытаний 10](#_Toc135453757)

[5.1.2 Программные средства, используемые во время испытаний 10](#_Toc135453758)

[5.2 Порядок проведения испытаний 10](#_Toc135453759)

[5.2.1 Стадии проведения испытаний 10](#_Toc135453760)

[5.2.2 Этапы проведения испытаний 11](#_Toc135453761)

[5.2.3 Содержание работ по этапам 11](#_Toc135453762)

[5.3 Характеристики, подлежащие оценке 12](#_Toc135453763)

[5.4 Условия проведения испытаний 13](#_Toc135453764)

[5.4.1 Общие условия 13](#_Toc135453765)

[5.4.2 Условия начала и завершения отдельных стадий испытаний 13](#_Toc135453766)

[5.4.3 Меры, обеспечивающие безопасность и безаварийность проведения испытаний 14](#_Toc135453767)

[5.4.4 Требования к персоналу, проводящему испытания 14](#_Toc135453768)

[5.5 Перечень работ, проводимых после завершения испытаний 14](#_Toc135453769)

[6 МЕТОДЫ ИСПЫТАНИЙ 16](#_Toc135453770)

[6.1 Методика проведения проверки комплектности программной документации 16](#_Toc135453771)

[6.2 Методика проведения проверки состава технических и программных средств 16](#_Toc135453772)

[6.3 Методика проведения модульного тестирования 16](#_Toc135453773)

[6.4 Методика проведения функционального тестирования 17](#_Toc135453774)

[6.5 Методика проведения автоматического тестирования 18](#_Toc135453775)

[6.6 Методика проведения тестирования производительности 19](#_Toc135453776)

[6.7 Методика проведения тестирования интерфейса 20](#_Toc135453777)

# 1 ОБЪЕКТ ИСПЫТАНИЙ

## 1.1 Наименование программного обеспечения

Объектом испытаний является программное обеспечение «Автогонки».

## 1.2 Область применения

Данный проект представляет собой приложение, на экране которого изображена дорога, на которой появляются машины и игроку следует их объезжать, чтобы не проиграть. Разрабатываемое приложение предназначено для развлечения и отдыха, а также для развития реакции, логического мышления, управления и координации движений у игроков. Приложение может применяться в качестве игры на разных типах персональных компьютеров

## 1.3 Обозначение испытуемого программного обеспечения

Условное обозначение испытуемого программного обеспечения (шифр) – «v1.0.0-АГ-ПК».

# 2 ЦЕЛЬ ИСПЫТАНИЙ

Цель проведения испытаний заключается в проверке соответствия разработанной программы функциональным и иным видам требований, изложенных в техническом задании и настоящем документе.

# 3 ТРЕБОВАНИЯ К ПРОГРАММЕ

## 3.1 Требования к отдельным модулям программы

### 3.1.1 Модуль генерации новых противников

К модулю генерации новых противников предъявляются следующие требования:

* новые противники появляются каждые 2 секунды;
* противники должны появляться на случайной свободной позиции в игровом пространстве.

### 3.1.2 Модуль перемещения игрока

К модулю перемещения игрока предъявляются следующие требования:

* игрок перемещается влево и вправо;
* игрок не может перемещаться за границы игрового пространства.

### 3.1.3 Модуль оперирования счётом

* при столкновении игрока и монетки должно начисляется одно очко;
* при запуске новой игры текущий счёт должен быть обнулён.

## 3.2 Функциональные требования

К программе предъявляются следующие функциональные требования:

* система должна предоставлять пользователю возможность заполнить форму с названием учётной записи и паролем для регистрации нового аккаунта, выполняя проверку введённых данных на валидность с последующим их сохранением в базе данных;
* система должна предоставлять пользователю возможность ознакомиться с информационной справкой, содержащей оправила игры;
* система должна предоставлять пользователю возможность начать новую игру, при этом должно быть обеспечено автоматическое создание игрового пространства со случайным расположением противников;
* система должна предоставлять пользователю возможность перемещать машину игрока на игровом пространстве, используя клавиши-стрелки на клавиатуре;
* система должна предоставлять пользователю возможность сбросить текущую игру и начать новую, при этом должно быть очищено текущее игровое пространство, а также сброшена статистика игры;
* система должна предоставлять пользователю возможность перемещаться между окнами приложения.

## 3.3 Требования к производительности

К производительности программы предъявляются следующие требования:

* приложение должно максимально эффективно использовать процессор, чтобы обеспечивать быстрое выполнение задач пользователей, ограничивая предельное потребление 10% от общих ресурсов в течение всего времени работы программы;
* приложение должно максимально эффективно использовать оперативную память устройства и предотвращать возможные ошибки из-за её дефицита, не превышая 100 Мб в течение всего времени работы программы;
* приложение должно максимально эффективно использовать графический процессор, чтобы обеспечивать быстрое выполнение задач пользователей, ограничивая предельное потребление 5% от общих ресурсов в течение всего времени работы программы.

## 3.4 Требования к интерфейсу

### 3.4.1 Требования к размещению пользовательского интерфейса

Пользовательский интерфейс программы должен представлять собой четыре окна:

1. окно регистрации пользователя с разрешением 525 x 517 пикселей;
2. окно игровой сессии с разрешением 525 x 517 пикселей;
3. окно общей игровой статистики с разрешением 525 x 517 пикселей.
4. окно с информацией по игре с разрешением 525 x 517 пикселей.

Все окна программы не должны масштабироваться и менять своё начальное разрешение, но их положение на экране может быть изменено.

### 3.4.2 Требования к наблюдаемости состояния системы

Во время игровой сессии в случае поражения система оповещает пользователя об этом соответствующим окном, после которого можно продолжить или закончить игру соответственно.

### 3.4.3 Требования к гибкости и эффективности использования

Программа должна предоставлять пользователю возможность выполнять следующие действия при помощи горячих клавиш:

* начало новой игры (клавиша «Enter»);
* перемещение машины игрока влево и вправо (клавиши «Left Arrow» и «Right Arrow»);
* выход из приложения (клавиша «Esc»).

### 3.4.4 Требования к справочной информации

Программа должна предоставлять пользователю доступ к справочной информации и инструкциям в окне игровой сессии при нажатии на кнопку с соответствующим названием, что поможет ознакомиться с правилами игры.

### 3.4.5 Требования к системе распознавания, диагностики и устранения ошибок

Сообщения об ошибках, возникающие при взаимодействии пользователя с системой, должны быть представлены не в виде кода ошибки, а в виде текста на естественном языке. Помимо этого, само сообщение должно конкретно указывать на ошибку, чтобы пользователь смог определить причины её возникновения и устранить их. Сам текст должен быть красного цвета, полужирного начертания и 18 размера.

### 3.4.6 Требования к предотвращению ошибок

Программа должна предупреждать пользователя о последствиях его действий и выполнять их только в случае подтверждения. Таким образом, действия начала новой игры, выхода из приложения или учётной записи должны сопровождаться всплывающим окном с подтверждением соответствующего действия.

# 4 ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

## 4.1 Состав программной документации, предъявляемой на испытания

Состав программной документации должен включать в себя:

* техническое задание;
* программу и методику испытаний;
* руководство пользователя.

## 4.2 Специальные требования

Специальные требования к программной документации не предъявляются.

# 5 СРЕДСТВА И ПОРЯДОК ИСПЫТАНИЙ

## 5.1 Средства испытаний

### 5.1.1 Технические средства, используемые во время испытаний

В состав технических средств, используемых во время испытаний, должен входить персональный компьютер, включающий в себя:

* оперативную память объёмом не менее 4 ГБ;
* процессор, с частотой не менее 3.0 ГГц;
* не менее 10 Гб свободного места на жестком диске.

### 5.1.2 Программные средства, используемые во время испытаний

Персональный компьютер, используемый во время проведения испытаний программы, должен быть оснащён следующим программным обеспечением:

* операционной системой Windows;
* программным пакетом .NET Framework;
* драйверами программного обеспечения Microsoft SQL Server для работы с базами данных.

Системные программные средства должны быть представлены лицензионной локализованной версией.

## 5.2 Порядок проведения испытаний

### 5.2.1 Стадии проведения испытаний

Испытания должны быть проведены в две стадии:

1. предварительная подготовка;
2. испытания.

### 5.2.2 Этапы проведения испытаний

Стадия «Предварительная подготовка» предназначена для знакомства с тестируемым объектом. Она должна включать в себя следующие этапы:

* программная документация;
* технические и программные средства.

Методики проведения проверок для этапов, входящих в перечень первой стадии испытаний, изложены в разделах 6.1 «Методика проведения проверки комплектности программной документации» и 6.2 «Методика проведения проверки состава технических и программных средств» соответственно.

Стадия «Испытания» выполняется с целью проверки соответствия программы установленным требованиям. Она должна включать в себя следующие этапы:

* модульное тестирование;
* функциональное тестирование;
* автоматическое тестирование;
* тестирование производительности;
* тестирование интерфейса.

Методики проведения проверок для этапов, входящих в перечень второй стадии испытаний, изложены в разделах 6.3 «Методика проведения модульного тестирования», 6.4 «Методика проведения функционального тестирования», 6.5 «Методика проведения автоматического тестирования», 6.6 «Методика проведения тестирования производительности» и 6.7 «Методика проведения тестирования интерфейса» соответственно.

### 5.2.3 Содержание работ по этапам

На этапе «Программная документация» должна быть проведена проверка комплектности программной документации в соответствии с разделом 4.1 «Состав программной документации, предъявляемой на испытания».

На этапе «Технические и программные средства» должна быть проведена проверка состава технических и программных средств в соответствии с разделом 5.1 «Средства испытаний».

На этапе «Модульное тестирование» проводится проверка минимально возможного для тестирования компонента в виде отдельного класса или функции в соответствии с разделом 3.1 «Требования к отдельным модулям программы».

Этап «Функциональное тестирование» проводится для проверки соответствия программного продукта его функциональным требованиям, определённым в разделе 3.2 «Функциональные требования».

На этапе «Автоматическое тестирование» проводится процесс верификации программного обеспечения, при котором отдельные методы и классы, определённые разделом 3.1 «Требования к отдельным модулям программы», выполняются автоматически при помощи инструментов для автоматизированного тестирования

Этап «Тестирование производительности» направлен на проверку скорости работы программы и её ресурсоёмкости в различных условиях нагрузки в соответствии с разделом 3.3 «Требования к производительности».

Этап «Тестирование интерфейса» осуществляется для проверки удобства использования интерфейса программы в соответствии с требованиями, определёнными в разделе 3.4 «Требования к интерфейсу».

## 5.3 Характеристики, подлежащие оценке

В ходе проведения испытаний оценке подлежат следующие характеристики:

1. комплектность программной документации;
2. состав технических и программных средств;
3. покрытие строк кода;
4. количество успешных тестов;
5. количество проваленных тестов;
6. процент исправленных дефектов;
7. время тестирования сборки.

## 5.4 Условия проведения испытаний

### 5.4.1 Общие условия

Испытания должны проводиться в нормальных условиях в соответствии с ГОСТ 22261-94. Условия проведения испытаний следующие:

* частота питающей электросети, Гц – 50 ± 0,5;
* напряжение питающей сети переменного тока, В – 220 ± 4.

### 5.4.2 Условия начала и завершения отдельных стадий испытаний

Необходимым и достаточным условием завершения первой стадии испытаний и начала второй стадии испытаний является успешное выполнение этапов, проводимых на первой стадии (см. раздел 5.2.2 «Этапы проведения испытаний»).

Условием завершения второй стадии испытаний является успешное выполнение этапов, проводимых на второй стадии испытаний (см. раздел 5.2.2 «Этапы проведения испытаний»).

### 5.4.3 Меры, обеспечивающие безопасность и безаварийность проведения испытаний

Для обеспечения безопасности при работе с техническими средствами необходимо соблюдать правила техники безопасности, которые предусмотрены при работе с электрооборудованием.

Организационно-технические мероприятия должны обеспечивать безаварийность и безопасность работы обслуживающего персонала при установке, настройке, испытаниях и эксплуатации системы.

### 5.4.4 Требования к персоналу, проводящему испытания

Персонал, проводящий испытания программного продукта, должен соответствовать следующим требованиям:

* опыт работы с компьютерами на уровне продвинутого пользователя, включая их настройку и обслуживание.
* наличие знаний в области тестирования программного обеспечения, включая умение писать тестовые сценарии и анализировать результаты тестирования;
* знание языка программирования C#, включая умение использовать инструменты тестирования, такие как отладчики и профилировщики;
* ориентироваться в функциональных возможностях программного продукта.

## 5.5 Перечень работ, проводимых после завершения испытаний

В случае успешного завершения испытаний должно быть подготовлено заключение о соответствии, в котором подводятся итоги испытаний и делается вывод о том, что объект испытания соответствует установленным требованиям.

В случае выявления несоответствия разработанной программы отдельным требованиям необходимо провести корректировку программы и программной документации по результатам испытаний.

# 6 МЕТОДЫ ИСПЫТАНИЙ

## 6.1 Методика проведения проверки комплектности программной документации

В ходе проверки комплектности программной документации сопоставляется состав документации с перечнем программной документации, приведенным в разделе 4.1 «Состав программной документации».

Проверка считается завершенной в случае соответствия состава и комплектности программной документации перечню документации, приведенному в указанном выше пункте.

## 6.2 Методика проведения проверки состава технических и программных средств

В ходе проверки сопоставляется состав и комплектность технических и программных средств с перечнями, приведенными в разделах 5.1.1 «Технические средства, используемые во время испытаний» и 5.1.2 «Программные средства, используемые во время испытаний» соответственно.

Проверка считается завершенной в случае соответствия состава и комплектности технических и программных средств перечням, приведённым в указанных выше пунктах.

## 6.3 Методика проведения модульного тестирования

Модульное тестирование проводится после завершения разработки отдельного модуля и его отладки, а также осуществляется разработчиком модуля. Перед проведением тестирования разработчик должен ознакомиться с техническим заданием на соответствующий модуль и требованиями к нему.

В ходе модульного тестирования должен быть составлен план тестирования, включающий тестовые случаи для проверки требований, приведённых в разделе 3.1 «Требования к отдельным модулям программы».

Проведение тестирования включает выполнение тестовых случаев не менее чем на двух наборах входных данных (корректный и некорректный), запись результатов тестирования и анализ полученных результатов. Результаты модульного тестирования документируются в отчёте, который включает описание тестовых случаев и результаты выполнения каждого тестового случая. При обнаружении ошибок должен быть сформирован отчёт об ошибках, включающий в себя описание дефектов и предложения по их устранению. Разработчик должен устранить выявленные дефекты и повторно протестировать модуль до их полного устранения.

Оценка качества модульного тестирования осуществляется на основе раздела 5.3 «Характеристики, подлежащие оценке» настоящего документа.

## 6.4 Методика проведения функционального тестирования

Функциональное тестирование проводится после завершения разработки программного продукта и его отладки, а также осуществляется специалистами, не участвовавшими в разработке программного продукта, но отвечающие требованиям, определённым в разделе 5.4.4 «Требования к персоналу, проводящему испытания». Перед проведением тестирования специалисты должны ознакомиться с техническим заданием на программный продукт и требованиями к нему.

В ходе функционального тестирования должен быть составлен план тестирования, включающий тестовые случаи для проверки требований, приведённых в разделе 3.2 «Функциональные требования».

Проведение тестирования включает выполнение тестовых случаев не менее чем на двух наборах входных данных (корректный и некорректный), запись результатов тестирования и анализ полученных результатов. Результаты функционального тестирования документируются в отчёте, который включает описание тестовых случаев и результаты выполнения каждого тестового случая. При обнаружении ошибок должен быть сформирован отчёт об ошибках, включающий в себя описание дефектов и предложения по их устранению. Разработчики должны устранить выявленные дефекты и повторно протестировать программный продукт до их полного устранения.

Оценка качества функционального тестирования осуществляется на основе раздела 5.3 «Характеристики, подлежащие оценке» настоящего документа.

## 6.5 Методика проведения автоматического тестирования

Автоматическое тестирование проводится после завершения разработки программного продукта и его отладки, а также осуществляется специалистами, не участвовавшими в разработке программного продукта, но отвечающие требованиям, определённым в разделе 5.4.4 «Требования к персоналу, проводящему испытания. Для проведения автоматического тестирования необходимо иметь набор тестовых скриптов и автоматизированный инструментарий для их выполнения.

В ходе автоматического тестирования должен быть составлен план тестирования, включающий набор тестовых скриптов для проверки требований, приведённых в разделе 3.1 «Требования к отдельным модулям программы».

Проведение тестирования включает запуск набора тестовых скриптов, запись результатов тестирования и анализ полученных результатов. Результаты автоматического тестирования документируются в отчёте, который включает описание тестовых скриптов и результаты выполнения каждого тестового скрипта. При обнаружении ошибок должен быть сформирован отчёт об ошибках, включающий в себя описание дефектов и предложения по их устранению. Разработчики должны устранить выявленные дефекты и повторно протестировать программный продукт до их полного устранения.

Оценка качества автоматического тестирования осуществляется на основе раздела 5.3 «Характеристики, подлежащие оценке» настоящего документа.

## 6.6 Методика проведения тестирования производительности

Тестирование производительности может проводиться на разных стадиях разработки программного продукта для выявления проблем производительности на ранних этапах, но обязательно проводится перед выпуском программы в эксплуатацию. Для проведения тестирования необходимо иметь специализированный инструмент «Performance Profiler» для измерения производительности программного продукта, а также набор тестовых скриптов.

В ходе тестирования производительности должен быть составлен план тестирования, включающий набор тестовых скриптов для проверки производительности программного продукта и требований, приведённых в разделе 3.3 «Требования к производительности».

Проведение тестирования включает запуск набора тестовых скриптов, запись результатов тестирования и анализ полученных результатов. Результаты тестирования производительности документируются в отчёте, который включает описание тестовых скриптов и результаты выполнения каждого тестового скрипта. При обнаружении узких мест в программе должен быть сформирован отчёт, включающий в себя описание проблемы и предложения по её оптимизации. Разработчики должны устранить выявленные дефекты и повторно протестировать программный продукт до их полного устранения.

Оценка качества тестирования производительности осуществляется на основе раздела 5.3 «Характеристики, подлежащие оценке» настоящего документа.

## 6.7 Методика проведения тестирования интерфейса

Тестирование интерфейса может проводиться на разных стадиях разработки программного продукта для выявления проблем интерфейса на ранних этапах, но обязательно проводится перед выпуском программы в эксплуатацию, а также осуществляется специалистами, не участвовавшими в разработке программного продукта и отвечающие требованиям, определённым в разделе 5.4.4 «Требования к персоналу, проводящему испытания».

В ходе тестирования интерфейса должен быть составлен план тестирования, включающий тестовые случаи для проверки удобства использования программного продукта и требований, приведённых в разделе 3.4 «Требования к интерфейсу».

Проведение тестирования включает выполнение тестовых случаев, запись результатов тестирования и анализ полученных результатов. Результаты функционального тестирования документируются в отчёте, который включает описание тестовых случаев и результаты выполнения каждого тестового случая. При обнаружении проблем должен быть сформирован отчёт, включающий в себя описание отклонений и предложения по их устранению. Разработчики должны устранить выявленные проблемы и повторно протестировать программный продукт до их полного устранения.

Оценка качества тестирования интерфейса осуществляется на основе раздела 5.3 «Характеристики, подлежащие оценке» настоящего документа.

ПРИЛОЖЕНИЕ 1

Модульное тестирование

Позитивный Test-Case

|  |  |  |
| --- | --- | --- |
| Действие | Ожидаемый результат | Фактический результат |
| 1. Открыть программу 2. В верхнем меню выбрать раздел «Сложность» 3. Нажать на кнопку «своя» 4. В открывшемся диалоговом окне ввести число 10 5. Нажать кнопку «ОК» 6. Открыть программу 7. В верхнем меню выбрать раздел «Сложность» 8. Нажать на кнопку «своя» 9. В открывшемся диалоговом окне ввести число 10,5 10. Нажать кнопку «ОК» | Программа корректно работает, в верхнем меню появилось значение введённой скорости | Программа корректно работает, в верхнем меню появилось значение введённой скорости |

Негативный Test-Case

|  |  |  |
| --- | --- | --- |
| Действие | Ожидаемый результат | Фактический результат |
| 1. Открыть программу 2. В верхнем меню выбрать раздел «Сложность» 3. Нажать на кнопку «своя» 4. В открывшемся диалоговом окне ввести число -10 5. Нажать кнопку «ОК» 6. Открыть программу 7. В верхнем меню выбрать раздел «Сложность» 8. Нажать на кнопку «своя» 9. В открывшемся диалоговом окне ввести слово «Скорость» 10. Нажать кнопку «ОК» 11. Открыть программу 12. В верхнем меню выбрать раздел «Сложность» 13. Нажать на кнопку «своя» 14. В открывшемся диалоговом окне ввести знаки .,!? 15. Нажать кнопку «ОК» | Появилось сообщение о некорректном вводе | * Сообщение о некорректном вводе не появилось * Ошибка System.FormatException * Сообщение о некорректном вводе не появилось |

Дефект репорт

|  |  |
| --- | --- |
| Короткое описание | Входная строка имела неверный формат |
| Проект | «Автогонки» |
| Компонент приложения | Метод my\_Click. Отвечает за ввод собственной сложности. |
| Номер версии | 1.0 |
| Серьёзность | Блокирующий |
| Приоритет | Высокий |
| Статус | Исправлен |
| Автор | Сова Сергей |
| Назначение | Сова Сергей |
| Окружение | |
| ОС/ Сервис Пак и т.д./ Браузер + версия/ … | ПО: Visual Studio 2017 |
| Описание | |
| Шаги воспроизведения | 1. Открыть программу 2. В верхнем меню выбрать раздел «Сложность» 3. Нажать на кнопку «своя» 4. В открывшемся диалоговом окне ввести любое значение, кроме положительного числа 5. Нажать кнопку «ОК» |
| Фактический результат | Ошибка System.FormatException |
| Ожидаемый результата | Появилось сообщение о некорректном вводе |
| Дополнение | |
| Прикрепленный файл |  |

|  |  |
| --- | --- |
| Короткое описание | Игра идёт в обратном направлении, то есть машинка едет назад, а не в перёд |
| Проект | «Автогонки» |
| Компонент приложения | Метод my\_Click. Отвечает за ввод собственной сложности. |
| Номер версии | 1.0 |
| Серьёзность | Блокирующий |
| Приоритет | Высокий |
| Статус | Исправлен |
| Автор | Сова Сергей |
| Назначение | Сова Сергей |
| Окружение | |
| ОС/ Сервис Пак и т.д./ Браузер + версия/ … | ПО: Visual Studio 2017 |
| Описание | |
| Шаги воспроизведения | 1. Открыть программу 2. В верхнем меню выбрать раздел «Сложность» 3. Нажать на кнопку «своя» 4. В открывшемся диалоговом окне ввести отрицательно число 5. Нажать кнопку «ОК» |
| Фактический результат | Игра идёт в обратном направлении, то есть машинка едет назад, а не в перёд |
| Ожидаемый результата | Появилось сообщение о некорректном вводе |
| Дополнение | |
| Прикрепленный файл |  |

Тестовые метрики

|  |  |
| --- | --- |
| Наименование метрики | Результат |
| Покрытие строк кода | 20% |
| Количество успешных тестов | 2 |
| Количество проваленных тестов | 3 |
| Процент исправленных дефектов | 100% |
| Время на тестирование сборки | 10 минут |

Комплект программной документации и состав средств соответствует всем требованиям, определённым в разделах 4.1 «Состав программной документации, предъявляемой на испытания» и 5.1 «Средства испытаний» соответственно

ПРИЛОЖЕНИЕ 2

Функциональное тестирование

Позитивный Test-Case

|  |  |  |
| --- | --- | --- |
| Действие | Ожидаемый результат | Фактический результат |
| 1. Открыть программу 2. В верхнем меню нажать кнопку «Старт» 3. Нажать на клавиатуре кнопку вправо 4. Нажать на клавиатуре кнопку влево 5. Открыть программу 6. В верхнем меню нажать кнопку «Старт» 7. Управляя машинкой, врезаться в другую машину 8. Открыть программу 9. В верхнем меню нажать кнопку «Сложность» 10. Нажать кнопку «Низкая» 11. Нажать кнопку «Старт» 12. Открыть программу 13. В верхнем меню нажать кнопку «Сложность» 14. Нажать кнопку «Средняя» 15. Нажать кнопку «Старт» 16. Открыть программу 17. В верхнем меню нажать кнопку «Сложность» 18. Нажать кнопку «Высокая» 19. Нажать кнопку «Старт» 20. Открыть программу 21. В верхнем меню нажать кнопку «Сложность» 22. Нажать кнопку «Своя» 23. В открывшемся диалоговом окне вписать собственную скорость 24. Нажать кнопку «ОК» 25. Нажать кнопку «Старт» | * Машинка переместилась вправо.   Машинка переместилась влево.   * Игра завершилась * Машинка двигается с низкой скоростью * Машинка двигается со средней скоростью * Машинка двигается с высокой скоростью * Машинка двигается с ведённой скоростью | * Машинка переместилась вправо.   Машинка переместилась влево   * Игра завершилась * Машинка двигается с низкой скоростью * Машинка двигается со средней скоростью * Машинка двигается с высокой скоростью * Машинка двигается с ведённой скоростью |

Негативный Test-Case

|  |  |  |
| --- | --- | --- |
| Действие | Ожидаемый результат | Фактический результат |
| 1. Открыть программу 2. В верхнем меню нажать на кнопку «Выход» 3. Открыть программу 4. В верхнем меню нажать кнопку «Старт» 5. Управляя машиной, собрать монету | * Программа завершила работу * Счётчик монет увеличился на единицу | * Программа продолжает работать * Счётчик монет не изменился |

Дефект репорт

|  |  |
| --- | --- |
| Короткое описание | При нажатии кнопки «Выход» программа продолжает работать |
| Проект | «Автогонки» |
| Компонент приложения | Метод Close\_Click. Отвечает за выход из приложения |
| Номер версии | 1.0 |
| Серьёзность | Блокирующий |
| Приоритет | Высокий |
| Статус | Исправлен |
| Автор | Сова Сергей |
| Назначение | Сова Сергей |
| Окружение | |
| ОС/ Сервис Пак и т.д./ Браузер + версия/ … | ПО: Visual Studio 2017 |
| Описание | |
| Шаги воспроизведения | 1. Открыть программу 2. В верхнем меню нажать на кнопку «Выход» |
| Фактический результат | Программа продолжает работать |
| Ожидаемый результата | Программа завершилась |
| Дополнение | |
| Прикрепленный файл |  |
| Короткое описание | Счётчик монет не изменяется при сборе монетки |
| Проект | «Автогонки» |
| Компонент приложения | Метод GameLoop. Отвечает за события, происходящие во время игры |
| Номер версии | 1.0 |
| Серьёзность | Блокирующий |
| Приоритет | Высокий |
| Статус | Исправлен |
| Автор | Сова Сергей |
| Назначение | Сова Сергей |
| Окружение | |
| ОС/ Сервис Пак и т.д./ Браузер + версия/ … | ПО: Visual Studio 2017 |
| Описание | |
| Шаги воспроизведения | 1. Открыть программу 2. В верхнем меню нажать кнопку «Старт» 3. Управляя машиной, собрать монету |
| Фактический результат | Счётчик монет не изменяется при сборе монетки |
| Ожидаемый результата | Счётчик монет увеличился на единицу |
| Дополнение | |
| Прикрепленный файл |  |

Тестовые метрики

|  |  |
| --- | --- |
| Наименование метрики | Результат |
| Покрытие строк кода | 15% |
| Количество успешных тестов | 6 |
| Количество проваленных тестов | 2 |
| Процент исправленных дефектов | 100% |
| Время на тестирование сборки | 27 минут |

Комплект программной документации и состав средств соответствует всем требованиям, определённым в разделах 4.1 «Состав программной документации, предъявляемой на испытания» и 5.1 «Средства испытаний» соответственно.

ПРИЛОЖЕНИЕ 3

Автоматическое тестирование

Позитивный Test-Case

|  |  |  |
| --- | --- | --- |
| Действие | Ожидаемый результат | Фактический результат |
| 1. Открыть программу 2. В верхнем меню нажать кнопку «Старт» 3. Дождаться появления диалогового окна «Конец игры». Нажать кнопку «ОК» 4. Нажать кнопку «Выход» 5. Открыть программу 6. В верхнем меню нажать кнопку «Сложность» 7. Нажать кнопку «Низкая» 8. Нажать кнопку «Старт» 9. Дождаться появления диалогового окна «Конец игры». Нажать кнопку «ОК» 10. В верхнем меню нажать кнопку «Сложность» 11. Нажать кнопку «Средняя» 12. Нажать кнопку «Старт» 13. Дождаться появления диалогового окна «Конец игры». Нажать кнопку «ОК» 14. В верхнем меню нажать кнопку «Сложность» 15. Нажать кнопку «Высокая» 16. Нажать кнопку «Старт» 17. Дождаться появления диалогового окна «Конец игры». Нажать кнопку «ОК» 18. Нажать кнопку «Выход» | * Игра запустилась, завершилась и закрылась * При увеличении уровня сложности скорость игры увеличивается | * Игра запустилась, завершилась и закрылась * При увеличении уровня сложности скорость игры увеличивается |

Негативный Test-Case

|  |  |  |
| --- | --- | --- |
| Действие | Ожидаемый результат | Фактический результат |
| 1. Открыть программу 2. Нажать кнопку «Старт» 3. Дождаться появления диалогового окна «Конец игры». Нажать кнопку «ОК» 4. В верхнем меню нажать кнопку «Сложность» 5. Нажать кнопку «x2» 6. Нажать кнопку «Старт» 7. Дождаться появления диалогового окна «Конец игры». Нажать кнопку «ОК» 8. Нажать кнопку «Выход» | * Скорость увеличилась в 2 раза | * Скорость не изменилась |

Дефект репорт

|  |  |
| --- | --- |
| Короткое описание | При нажатии кнопки «x2» скорость не изменяется |
| Проект | «Автогонки» |
| Компонент приложения | Метод x2\_Click. Отвечает за увеличение скорости в 2 раза |
| Номер версии | 1.0 |
| Серьёзность | Незначительный |
| Приоритет | Средний |
| Статус | Исправлен |
| Автор | Сова Сергей |
| Назначение | Сова Сергей |
| Окружение | |
| ОС/ Сервис Пак и т.д./ Браузер + версия/ … | ПО: Visual Studio 2017 |
| Описание | |
| Шаги воспроизведения | 1. Открыть программу 2. В верхнем меню нажать кнопку «Сложность» 3. Нажать кнопку «x2» 4. Нажать кнопку «Старт» |
| Фактический результат | Скорость не изменилась |
| Ожидаемый результата | Скорость увеличилась в 2 раза |
| Дополнение | |
| Прикрепленный файл |  |

Тестовые метрики

|  |  |
| --- | --- |
| Наименование метрики | Результат |
| Покрытие строк кода | 100% |
| Количество успешных тестов | 2 |
| Количество проваленных тестов | 1 |
| Процент исправленных дефектов | 100% |
| Время на тестирование сборки | 41 минут |

Комплект программной документации и состав средств соответствует всем требованиям, определённым в разделах 4.1 «Состав программной документации, предъявляемой на испытания» и 5.1 «Средства испытаний» соответственно

Код разработанного скрипта



Рис. 1 Код первого позитивного Test-Case



Рис. 2 Код второго позитивного Test-Case



Рис. 3 Код негативного Test-Case

ПРИЛОЖЕНИЕ 4

Тестирование производительности

Параметры аппаратуры и программного окружения

|  |  |
| --- | --- |
| Операционная система | «Microsoft Windows 11 Домашняя» на базе x64 |
| Процессор | 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz |
| Графический процессор | Intel(R) Iris(R) Xe Graphics |
| Материнская плата | HUAWEI NBD-WXX9-PCB-B4 |
| Оперативная память | SSD 8 Gb 1600 MHz |
| Жёсткий диск | Samsung ATA Device 512 Gb |
| Программное окружение | Visual Studio 2022 |

|  |  |  |  |
| --- | --- | --- | --- |
| Описание | Порядок действий | Ожидаемый результат | Фактический результат |
| Использование процессора | 1. Во вкладке «Отладка» выбрать пункт «Начать отладку» для запуска программы в режиме отладки.  2. Во вкладке «Отладка» выбрать пункт «Профилировщик производительности», в пункте «ЦП» проанализировать использование процессора на протяжении использования программы. | Нагрузка процессора не превышает 10% процентов доступной мощности | Нагрузка процессора не превышает 5% процентов доступной мощности. |
| Использование памяти | 1. Во вкладке «Отладка» выбрать пункт «Начать отладку» для запуска программы в режиме отладки.  2. Во вкладке «Отладка» выбрать пункт «Профилировщик производительности», в пункте «Память процессора» проанализировать использование памяти на протяжении использования программы. | Используемый объем памяти стабилен. Стандартная нагрузка – 100 МБ. | Используемый объем памяти стабилен. Стандартная нагрузка – 70 МБ. |
| Использование графического процессора | 1. Во вкладке «Отладка» выбрать пункт «Начать отладку» для запуска программы в режиме отладки.  2. Во вкладке «Отладка» выбрать пункт «Профилировщик производительности», в пункте «Использование GPU» проанализировать использование графического процессора на протяжении использования программы. | Нагрузка процессора не превышает 5% процентов доступной мощности | Нагрузка процессора не превышает 3,3% процентов доступной мощности |

Результаты тестирования производительности

1. Использование процессора. Максимальное значение 5%. Минимальное значение 0%. Среднее значение 2%.

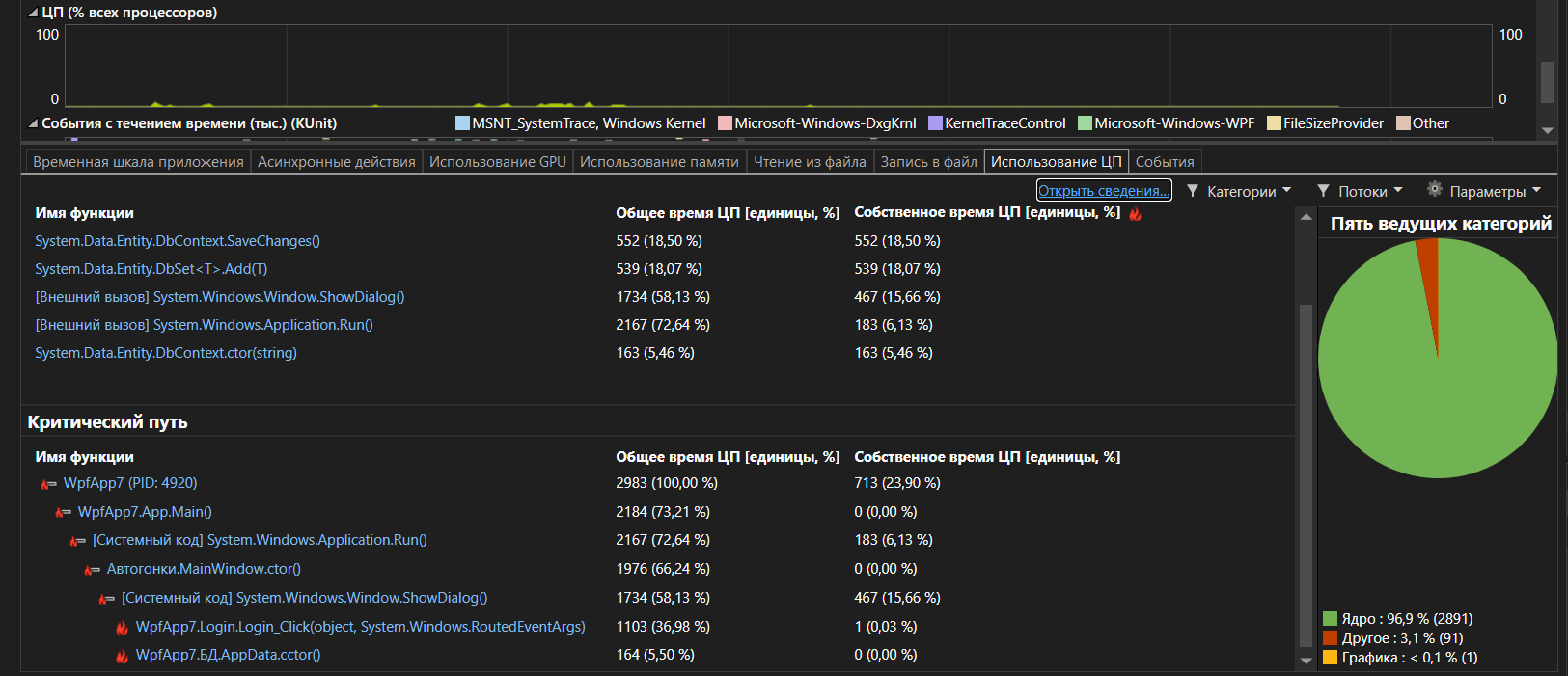


Рис. 1. График использования центрального процессора при прохождении тестового случая

1. Использование памяти. Максимальное значение 67 МБ. Минимальное значение 585 КБ. Среднее значение 66 МБ (рисунок 2).

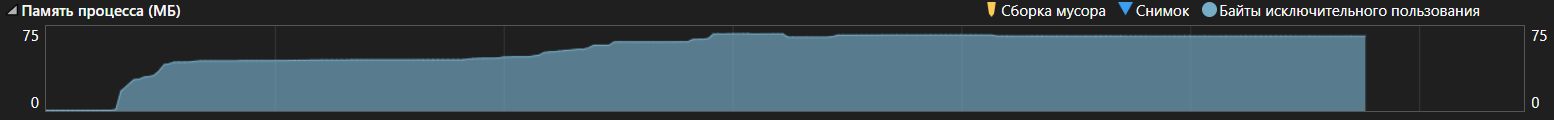


Рис. 2. График использования памяти при прохождении тестового случая

1. Использование графического процессора. Максимальное значение 3,3%. Минимальное значение 0,7%. Среднее значение 1,2% (рисунок 3).

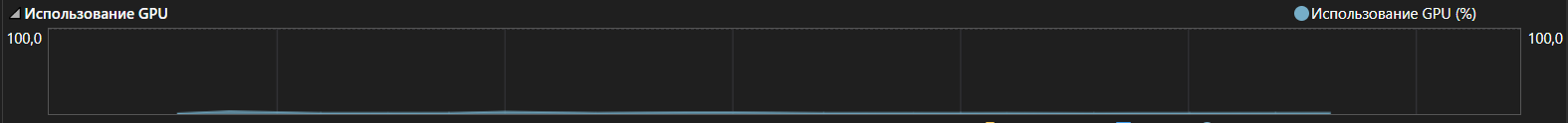


Рис. 3 График использования графического процессора при прохождении тестового случая

Вывод

После проведенных испытаний на производительность можно сказать, что разработанная система является достаточно производительной и не ресурсоёмкой. Ошибок обнаружено не было. Таким образом, система является надежной и вполне готовой к использованию.

ПРИЛОЖЕНИЕ 5

Тестирование интерфейса

|  |  |  |  |
| --- | --- | --- | --- |
| Название критерия | Описание | Оценка разработчика | Оценка эксперта |
| Наблюдаемость состояния системы | Система всегда должна оповещать пользователя о том, что она в данный момент делает, причем через разумные промежутки времени. | 9 | 7 |
| Соотнесение с реальным миром | Терминология, использованная в интерфейсе системы должна соотноситься с пользовательским миром, т.е. это должна быть терминология проблемной области пользователя, а не техническая терминология. | 10 | 7 |
| Пользовательское управление и свобода действий | Необходимо предоставлять четко определенный аварийный выход, при помощи которого можно вернуться к предыдущему нормальному состоянию. К таким "аварийным выходам" относятся, например, функции отката и обратного отката. | 5 | 7 |
| Целостность и стандарты | Для обозначения одних и тех же объектов, ситуаций и действий должны использоваться одинаковые слова во всех частях интерфейса. Более того, терминология сообщений в пользовательском интерфейсе должна учитывать соглашения конкретной платформы. | 9 | 7 |
| Помощь пользователям в распознавании, диагностике и устранении ошибок | Сообщение об ошибках должны быть написаны на естественном языке, а не заменяться кодами ошибок. Сообщения об ошибках должны четко определять суть возникшей проблемы и предлагать ее конструктивное решение. | 7 | 7 |
| Предотвращение ошибок | Продуманный дизайн пользовательского интерфейса, предотвращающий появление ошибок пользователя, всегда лучше хорошо продуманных сообщений об ошибках. | 7 | 7 |
| Распознавание, а не вспоминание | При создании интерфейса необходимо минимизировать нагрузку на память пользователя, делая объекты, действия и опции ясными, доступными и явно видимыми. Пользователь не должен запоминать информацию при переходе от одного диалогового окна к другому. Во всех необходимых местах должны быть доступны контекстные инструкции по использованию интерфейса. | 7 | 8 |
| Гибкость и эффективность использования | В интерфейсе должны быть предусмотрены горячие клавиши -они часто значительно ускоряют работу опытного пользователя. | 10 | 7 |
| Эстетичный и минимально необходимый дизайн | Окна не должны содержать не относящуюся к делу или редко используемую информацию. Каждый интерфейсный элемент, содержащий бесполезную информацию, играет роль информационного шума и отвлекает пользователя от действительно полезных интерфейсных элементов. | 5 | 4 |
| Помощь и документация | Информация в документации должна быть структурирована таким образом, чтобы пользователь мог легко найти нужный раздел, посвященный решаемой им задаче. Каждый такой ориентированный на конкретную задачу раздел должен помимо общей информации содержать пошаговые руководства по выполнению задачи и не должен быть слишком длинным. | 8 | 10 |

Тест-кейсы

|  |  |  |  |
| --- | --- | --- | --- |
| №  п/п | Действие | Реакция системы | Результат |
| 1 | Запустить программу «Автогонки» | Откроется окно регистрации пользователя с незаполненными полями | Фактическая реакция системы соответствует ожидаемому результату |
| 2 | Ввести в поле «Логин» значение «user», в поле «Пароль» - значение «123» и нажать на кнопку «Зарегистрироваться» | Высветится надпись «Вы зарегистрировались и откроется окно игровой сессии. | Фактическая реакция системы соответствует ожидаемому  результату |

|  |  |  |  |
| --- | --- | --- | --- |
| №  п/п | Действие | Реакция системы | Результат |
| 1 | Запустить программу «Автогонки» | Откроется окно регистрации пользователя с незаполненными полями | Фактическая реакция системы соответствует ожидаемому результату |
| 2 | Ввести в поле «Логин» значение «user» и нажать на кнопку «Зарегистрироваться» | Высветится надпись «Ошибка при вводе логина». | Фактическая реакция системы соответствует ожидаемому результату |
| 3 | Ввести в поле «Пароль» значение «123» и нажать на кнопку «Зарегистрироваться» | Высветится надпись «Ошибка при вводе пароля». | Фактическая реакция системы соответствует ожидаемому  результату |
| 4 | Ввести в поле «Логин» значение «user», в поле «Пароль» - значение «123», в и нажать на кнопку «Зарегистрироваться» | Высветится надпись «Вы зарегистрировались» и откроется окно игровой сессии. | Фактическая реакция системы соответствует ожидаемому результату |

|  |  |  |  |
| --- | --- | --- | --- |
| №  п/п | Действие | Реакция системы | Результат |
| 1 | Запустить программу «Автогонки» | Откроется окно регистрации пользователя с незаполненными полями | Фактическая реакция системы соответствует ожидаемому  результату |
| 2 | Нажать на кнопку «Об игре» | Откроется окно с правилами игры | Фактическая реакция системы соответствует ожидаемому  результату |

|  |  |  |  |
| --- | --- | --- | --- |
| №  п/п | Действие | Реакция системы | Результат |
| 1 | Запустить программу «Автогонки» | Откроется окно регистрации пользователя с незаполненными полями | Фактическая реакция системы соответствует ожидаемому  результату |
| 2 | Ввести в поле «Логин» значение «user», в поле «Пароль» - значение «123» и нажать на кнопку «Зарегистрироваться» | Высветится надпись «Вы зарегистрировались» и откроется окно игровой сессии. | Фактическая реакция системы соответствует ожидаемому  результату |
| 3 | Нажать на кнопку «Данные» | Откроется окно со всей информацией всех игровых сеансов | Фактическая реакция системы соответствует ожидаемому  результату |