|  |  |  |
| --- | --- | --- |
| СОГЛАСОВАНО |  | УТВЕРЖДАЮ |
| Преподаватель по анализу и разработке ТЗ  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ржаникова Е.Д.  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г. |  | Руководитель ОП  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Чистяков Г.А.  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г. |

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

на разработку

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_игры «Don’t Crash The Car»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |
| --- | --- | --- |
| СОГЛАСОВАНО |  | СОГЛАСОВАНО |
| Студент колледжа ВятГУ  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Обухов И.Н.  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г. |  | Преподаватель по учебной практике  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Крутиков А.К.  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г. |
|  |  | СОГЛАСОВАНО |
|  |  | Преподаватель по внедрению ИС  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Самоделкин П.А.  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г. |

2024

Содержание

[Введение 2](#_Toc163866704)

[1 Термины и определения 3](#_Toc163866705)

[2 Перечень сокращений 4](#_Toc163866706)

[3 Основные сведения о разработке 5](#_Toc163866707)

[3.1 Наименование разработки 5](#_Toc163866708)

[3.2 Цель и задачи 5](#_Toc163866709)

[3.3 Сведения об участниках разработки 5](#_Toc163866710)

[3.4 Сроки разработки 6](#_Toc163866711)

[3.5 Назначение разработки 6](#_Toc163866712)

[4 Описание предметной области 7](#_Toc163866713)

[5 Требования к результатам разработки 10](#_Toc163866714)

[5.1 Правила игры 10](#_Toc163866715)

[5.2 Требования к функциям 10](#_Toc163866716)

[5.3 Требования к показателям назначения 10](#_Toc163866717)

[5.4 Требования к пользовательскому интерфейсу 11](#_Toc163866718)

[5.5 Требования к видам обеспечения 13](#_Toc163866719)

[5.5.1 Требования к математическому обеспечению 13](#_Toc163866720)

[5.5.2 Требования к информационному обеспечению 13](#_Toc163866721)

[5.5.3 Требования к метрологическому обеспечению 14](#_Toc163866722)

[5.5.4 Требования к техническому обеспечению 14](#_Toc163866723)

[5.6 Требования к надежности 14](#_Toc163866724)

[5.7 Требования к безопасности 14](#_Toc163866725)

[5.8 Требования к патентной чистоте 14](#_Toc163866726)

[5.9 Требования к перспективам развития 15](#_Toc163866727)

[6 Состав и содержание работ 16](#_Toc163866728)

[7 Требования к документированию 17](#_Toc163866729)

[8 Требования к приемо-сдаточным процедурам 18](#_Toc163866730)

# Введение

Данный документ является техническим заданием для игрового приложения «Don’t Crash The Car», в котором описаны общие сведения о разработке, описание предметной области, требования к системе, функциям.

Документ предназначен для:

* разработчика приложения. Документ необходим разработчику с целью определения требований заказчика к проекту: описание функциональных требований, особенности архитектуры, используемые инструменты, технологии, сроки сдачи проекта и объем работы. Опираясь на техническое задание, разработчик исполняет конкретно поставленные задачи;
* членов приёмно-сдаточной комиссии. Техническое задание используется приемно-сдаточной комиссией с целью оценки на соответствие готового продукта заранее утвержденным требованиям. Приемно-сдаточная комиссия использует документ для того, чтобы проверить, что все функциональные, технические и дизайнерские требования были удовлетворены;
* заказчика. Техническое задание необходимо заказчику с целью определения требований к проекту и контроля над процессом его создания. Оно помогает заказчику понять, что он получит в результате работы и какие функции и возможности будут доступны в готовом продукте. Техническое задание позволяет заказчику убедиться, что проект соответствует его требованиям и ожиданиям, и что работы выполняются в соответствии с установленным бюджетом и графиком.

# Термины и определения

Документ содержит следующие термины и определения:

* интерфейс – набор инструментов, который позволяет пользователю взаимодействовать с программой;
* GUI ((graphical user interface) графический интерфейс пользователя) - система средств для взаимодействия пользователя с электронными устройствами, основанная на представлении всех доступных пользователю системных объектов и функций в виде графических компонентов экрана;
* IDE ((Integrated Development Environment) интегрированная среда разработки) - это программное приложение, которое помогает программистам эффективно разрабатывать программный код. Оно повышает производительность разработчиков, объединяя такие возможности, как редактирование, создание, тестирование и упаковка программного обеспечения в простом для использования приложении;
* UX ((User Experience) пользовательский опыт) - это область веб-дизайна и разработки приложений, которая отвечает за общее впечатление пользователя от продукта, его удобство использования и доступность. UI отвечает за внешний вид продукта, включая цветовую палитру, шрифты, расположение элементов на экране и другие аспекты, которые влияют на визуальное восприятие пользователей;
* UI ((User Interface) пользовательский интерфейс) - это область веб-дизайна и разработки приложений, которая отвечает за внешний вид продукта, включая цветовую палитру, шрифты, расположение элементов на экране и другие аспекты, которые влияют на визуальное восприятие пользователей.

# Перечень сокращений

Документ содержит следующие аббревиатуры и сокращения:

* УП – учебная практика;
* ГОСТ – государственный стандарт;
* СТП – стандарт предъприятия;
* ЕСПД – единая система программной документации.

# Основные сведения о разработке

В данном разделе настоящего технического задания описываются основные сведения о разработке игры «Don’t Crash The Car».

## Наименование разработки

Наименованием данной разработки является «Don’t Crash The Car».

## Цель и задачи

Целью разработки является создание игрового приложения «Don’t Crash The Car» в соответствии с требованиями и ограничениями, описанными в данном документе.

Для достижения поставленной цели необходимо выполнить следующие задачи:

* определить требования к приложению;
* спроектировать архитектуру приложения;
* разработать пользовательский интерфейс;
* реализовать логику появления машинок противника;
* реализовать логику счётчика очков;
* протестировать функциональность;
* отладка приложения;
* оптимизировать приложение;
* написать пакет необходимой информации;
* приемно-сдаточные испытания.

## Сведения об участниках разработки

Исполнителем является студент Колледжа ФГБОУ ВО «Вятский государственный университет» учебной группы ИСПк-203-52-00 Обухов Илья Николаевич.

Заказчиком является коллектив преподавателей Колледжа ФГБОУ ВО «Вятский государственный университет» в составе:

* Крутиков Александр Константинович (Преподаватель УП 05);
* Ржаникова Елена Дмитриевна (Преподаватель МДК 05.05);
* Самоделкин Павел Андреевич (Преподаватель МДК 06.01);
* Чистяков Геннадий Андреевич (Директор ИМИС ВятГУ).

## Сроки разработки

Разработка программного продукта должна быть осуществлена с 15.01.2024 по 19.06.2024.

## Назначение разработки

Функциональным назначением программного продукта является:

* развитие умений в сфере UX/UI;
* улучшение навыков программирования на языке Python;
* получение навыков работы с объектно-ориентированной парадигмой программирования;
* получение навыков разработки программного обеспечения и написание документации к нему.

Эксплуатационное назначение: продукт должен использоваться пользователем для отдыха и развлечения.

# Описание предметной области

Игры – это неотъемлемая часть нашей культуры, и термин «игра» часто используется в контексте детской деятельности. Через игры мы развиваем восприятие мира, улучшаем мышление и развиваем творческое воображение.

Автомобильные игры - это жанр компьютерных игр, в котором игрок управляет различными видами автомобилей, выполняя различные задачи или соревнуясь с другими игроками или искусственным интеллектом. Автомобильные игры могут быть реалистичными симуляторами вождения или аркадными гоночными играми, а также включать элементы экшена, стратегии или квестов. Этот жанр пользуется популярностью среди любителей автомобилей и скорости, а также тех, кто ценит динамичный игровой процесс.

«Don't Crash The Car» – это игровое приложение, в котором игрок на транспортном средстве должен маневрировать и обходить других участников движения, чтобы выполнить определенные задачи в условиях игры (набрать как можно больше очков и не столкнуться).

Игра предназначена для развлечения, тренировки реакции и координации движений. Она помогает игрокам развивать навыки быстрого принятия решений и улучшать свою реакцию.

На данный момент существуют следующие аналоги игрового приложения «Don't Crash The Car»:

**Аналог №1**

Название: «Rival Rush».

Ссылка: <https://logicplays.com/tetris/436-rival-rush.html>

Рассмотрим внешний вид аналога №1 (рис. 1):

Изображение выглядит как снимок экрана, текст

Автоматически созданное описание

Рисунок 1 – Внешний вид приложения

**Rival Rush** – это аркадная игра, в которой задачей является достичь финиша без аварий, уклоняясь от других игроков и обгоняя их. Сложность игры заключена в противниках дурного характера, которые готовы пожертвовать собой, лишь бы не дать вам дойти до финиша, а потому подрезают вас, вынуждая столкнуться.

К преимуществу аналога можно отнести проработанные детали внешнего вида персонажа.

К недостаткам можно отнести:

* отсутствие счётчика очков;
* работа приложения только при наличии выхода в Интернет.

**Аналог №2**

Название: «Tetro Car».

Ссылка: <http://gamva.ru/gonka-v-tetrise/>

Рассмотрим внешний вид аналога №3 (рис. 3):

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Рисунок 2 – Внешний вид приложения

**Tetro Car** - это аркадная игра с пиксельной графикой, в которой игрок управляет машиной, преодолевая препятствия. В Tetro Car нужно попытаться пройти уровень, избегая столкновений.

Преимуществом аналога является наличие счётчика очков.

Недостатками приложения являются:

* аналог доступен только на английском языке;
* работа только при наличии выхода в Интернет.

**Аналог №3**

Название: «Ретро Рейс».

Ссылка: <https://yandex.ru/games/app/214091>

Интерфейс аналога №3 (рис. 3):

Изображение выглядит как снимок экрана

Автоматически созданное описание

Рисунок 3 – Внешний вид приложения

**Ретро Рейс** - это аркадная гоночная игра с ретро-стилизацией, в которой игрок управляет автомобилем, преодолевая различные препятствия и собирая бонусы на пиксельных уровнях.

К преимуществам можем отнести:

* наличие счётчика очков
* наличие главного меню.

Недостатками приложения являются:

* отсутствие музыкального сопровождения
* отсутствие возможности управлять транспортным средством вперед и назад;
* аналог работает только при наличии выхода в Интернет.

Из анализа аналогов можно сделать вывод, что необходимо разработать такое приложение, в котором будут реализованы следующие возможности:

* отображение счётчика очков;
* наличие музыкального сопровождения;
* доступ к приложению без выхода в Интернет;
* игровое приложение должно быть на русском языке.

# Требования к результатам разработки

Результаты настоящей разработки должны соответствовать требованиям, которые содержатся в настоящем разделе.

## Правила игры

В игре присутствуют два типа машинок - машинка игрока и вражеские машинки, управляемые компьютером.

Игроку необходимо управлять машинкой, которая движется по трассе. Это единственный игровой объект, которым может управлять игрок. Управление осуществляется с помощью кнопок на клавиатуре, позволяющих машинке двигаться влево, вправо, вперед или назад по трассе.

Очки начисляются по мере продвижения машинки игрока по трассе. Чем дольше игрок избегает столкновений, тем больше очков он зарабатывает.

Цель игры - набрать как можно больше очков, избегая столкновений с машинами компьютера.

Игровой процесс завершается, когда машинка игрока сталкивается с вражеской машинкой. В конце игры отображается количество набранных очков.

Игрок может стремиться побить свой личный рекорд, набрав как можно больше очков в каждой новой игре.

## Требования к функциям

Игра должна обеспечивать следующие функции:

* начать игру при помощи нажатия на функциональную кнопку, отвечающую за запуск игры;
* перемещение машинки по игровому полю при помощи нажатия кнопок на клавиатуре;
* отображение счётчика очков;
* возможность включения, выключения фоновой музыки;
* возможность приостановить игру и возобновить её позже;
* отображение лучшего счета игрока;
* возможность перехода в главное меню после проигрыша;
* закрыть игру, путём нажатия на кнопку «Выход».

## Требования к показателям назначения

Требования к показателям назначения не предъявляются.

## Требования к пользовательскому интерфейсу

Пользовательский интерфейс настоящей разработки должен соответствовать требованиям, представленным в настоящем подразделе.

Меню игры «Don’t Crash The Car» имеет 3 функциональные кнопки (см. рис. 4): «Играть», «Музыка вкл/выкл», «Выход». В середине окна указывается название игры.

При нажатии функциональной кнопки «Играть» запустится игровой процесс, управляемый пользователем.

При нажатии функциональной кнопки «Музыка вкл/выкл» пользователь включит или выключит внутриигровое музыкальное сопровождение.

При нажатии функциональной кнопки «Выход» пользователь закроет игровое приложение.

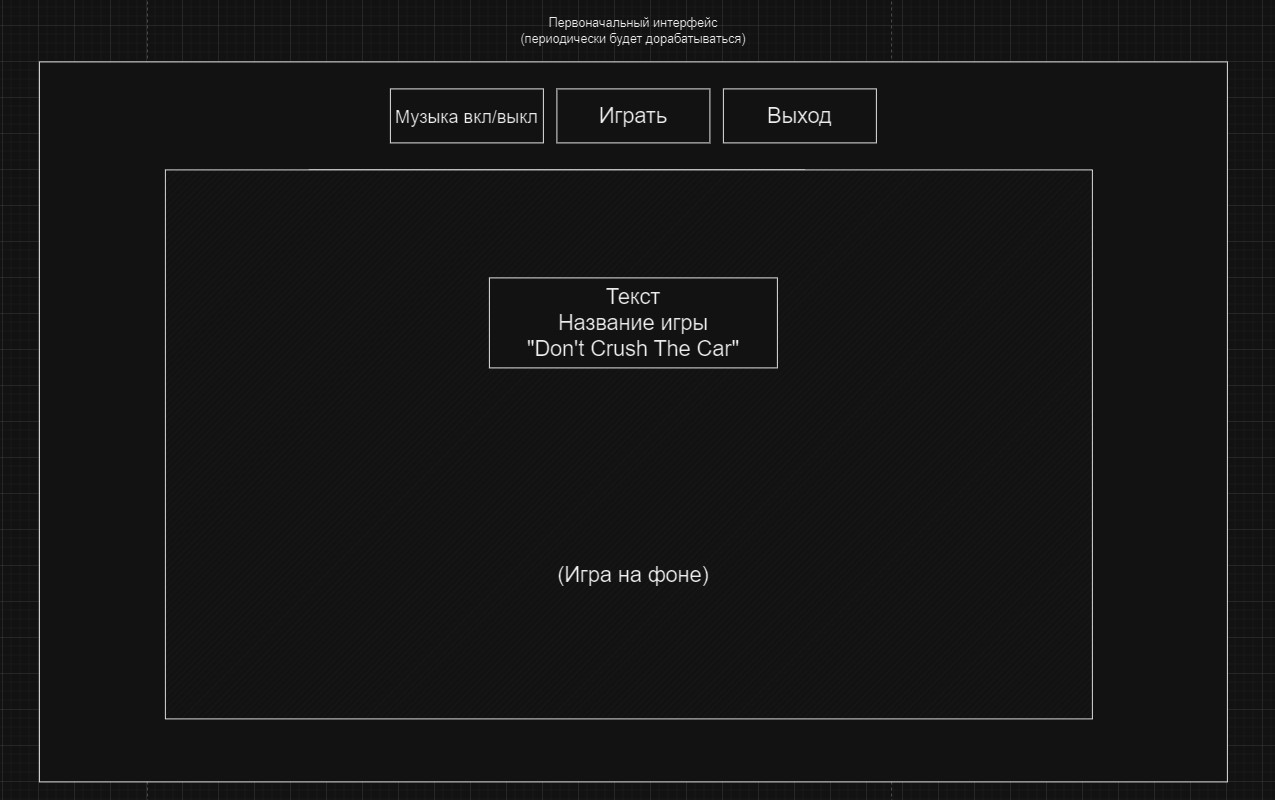


Рисунок 4 – Прототип пользовательского интерфейса

На рисунке 5 представлен прототип игрового процесса, состоящий из следующих элементов: машина игрока, машины компьютера, счётчик очков и функциональные кнопки, позволяющие пользователю включить или выключить музыку, выйти из игрового приложения или выйти в главное меню.

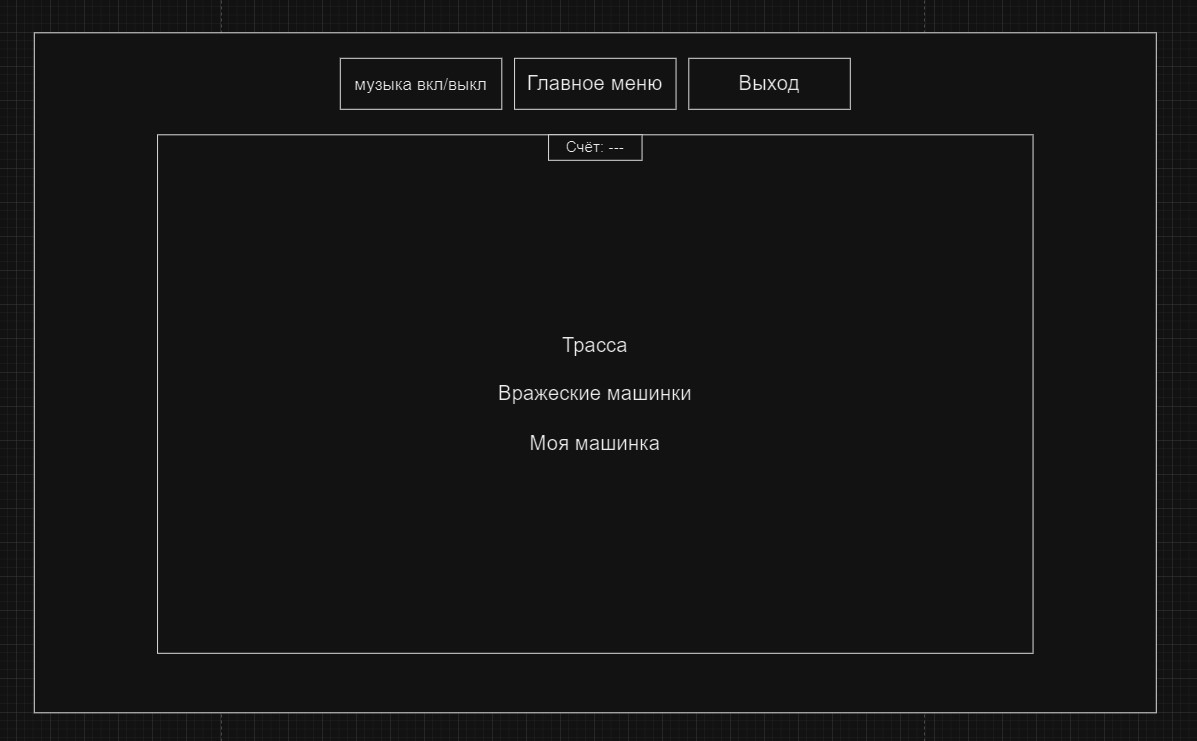


Рисунок 5 – Прототип игрового окна

На рисунке 6 представлен прототип информирующего окна, появляющееся при поражении. Данное окно содержит пояснительный текст о том, сколько очков набрал пользователь, функциональную кнопку «Музыка вкл/выкл», функциональную кнопку «Главное меню», а также функциональную кнопку «Выход».

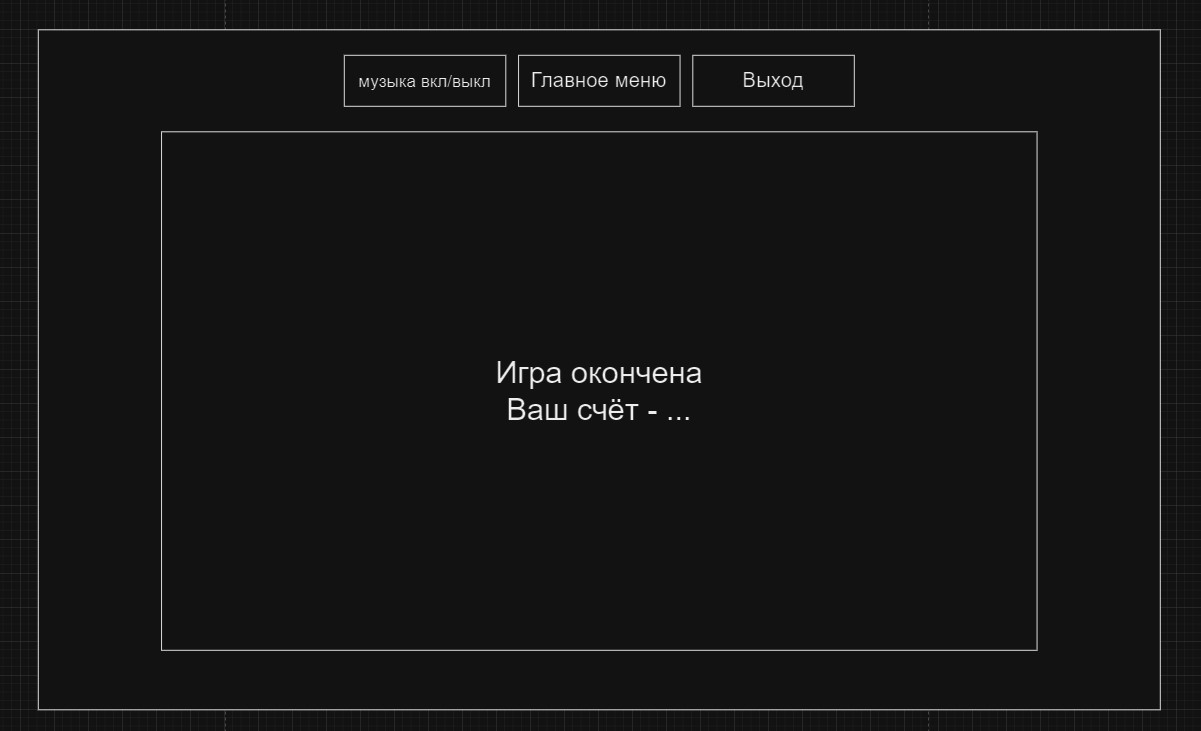


Рисунок 6 – Прототип окна при проигрыше

## Требования к видам обеспечения

В данном подразделе содержатся требования к видам обеспечения результата настоящей разработки.

### Требования к математическому обеспечению

Требование к математическому обеспечению для игры “Don’t Crash The Car” включает:

Создание противников: для определения координат нового противника используется генератор случайных чисел. Он также должен проверять, чтобы новые координаты не пересекались с координатами других машин, для этого используются абсолютные значения для расчета разницы между координатами.

Расчет очков: формула для расчета очков выглядит следующим образом:

новый счет = текущий счет + очки, где:

* новый счет - это обновленное значение счета после добавления очков;
* текущий счет - это текущее значение счета до добавления очков;
* очки - это количество очков, которые нужно добавить к текущему счету.

После расчета нового счета, он отображается на метке счета на экране.

### Требования к информационному обеспечению

Игровое приложение «Don’t Crash The Car» должно соответствовать следующим требованиям:

* графический интерфейс пользователя: нужно разработать интуитивно понятный графический интерфейс, который будет использоваться для игры;
* управление игрой: оптимальные способы управления игрой с помощью компьютерной мыши или тачпада;

#### Требования к форматам хранения данных

Требования к форматам хранения данных не предъявляются.

#### Требования к лингвистическому обеспечению

Игровое приложение «Don’t Crash The Car» должно предусматривать полностью русскую локализацию.

### Требования к метрологическому обеспечению

Требования к метрологическому обеспечению не предъявляются.

### Требования к техническому обеспечению

Приложение должно запускаться на персональных компьютерах, обладающих следующими минимальными характеристиками:

* программное обеспечение: операционная система Windows 10;
* устройство вывода: монитор;
* устройство ввода: компьютерная мышь.

## Требования к надежности

Игровое приложение «Don’t Crash The Car» должно соответствовать следующим требованиям:

* каждый элемент интерфейса должен быть рабочим, срабатывать при нажатии, что бы не возникало проблем с использованием;
* интерфейс должен четко отображать все компоненты приложения на экране;
* регулярные тестирования приложения на выявление ошибок и их исправление для гарантирования стабильного использования;
* организация бесперебойного питания технических средств;
* регулярное выполнение рекомендаций Министерства труда и социального развития РФ, изложенных в Постановлении от 23 июля 1998 г. «Об утверждении  
  межотраслевых типовых норм времени на работы по сервисному обслуживанию  
  ПЭВМ и оргтехники и сопровождению программных средств».

## Требования к безопасности

В данной разработке требования к безопасности не предъявляются.

## Требования к патентной чистоте

При разработке игрового приложения «Don’t Crash The Car» должны быть выполнены следующие требования к патентной чистоте:

* все элементы игрового приложения «Don’t Crash The Car» (правила, дизайн, названия) должны быть уникальными и не нарушать никакие патентные права других компаний и приложений;
* игровое приложение «Don’t Crash The Car» не должно содержать элементы нарушающие авторские права или патентные права, такие как изображения, символы, названия или другой контент, принадлежащие другим компаниям или разработчикам;
* все компоненты, используемые в игровом приложении «Don’t Crash The Car», должны быть свободны от патентных споров или интеллектуальных прав, относящихся к другим играм или программным продуктам.

## Требования к перспективам развития

Игровое приложение «Don’t Crash The Car» должно соответствовать следующим требованиям:

* улучшение GUI – улучшение графического интерфейса повысит привлекательность приложения для пользователей, усилит её имидж;
* расширение функциональных требований – добавление новых функций и возможностей приложения.

# Состав и содержание работ

В рамках разработки программного продукта в соответствии с настоящим документом необходимо выполнить перечень работ, представленный в таблице 1.

Таблица 1 – Состав и содержание работ настоящей разработки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № этапа | Наименование этапа | Длительность | Состав работ | Результат |
| 1 | Настройка рабочего окружения | 1 неделя | В ходе работ оборудование должно быть подготовлено к написанию кода программы. | Рабочее место готово к написанию кода программы. |
| 2 | Написание кода программы | 2 месяца | Написать код на выбранном языке программирования. Создать игровое поле и интерфейс. | Готовая программа. |
| 3 | Тестирование | 1 месяц | Протестировать приложение на работоспособность и производительность. | Список недоработок. |
| 4 | Доработка программы | 2 недели | Исправление недочетов. | Недоработки устранены. |
| 5 | Подготовка эксплуатационной документации | 2 недели | Должно быть написано руководство пользователя. | Руководство пользователя написано. |
| 6 | Приемо-сдаточные испытания | 2 недели | Сдача результатов работ комиссии, проведение испытаний результатов разработки и занесение результатов в ведомость | Оценка в ведомости. |

# Требования к документированию

Состав программной документации должен включать в себя:

* техническое задание, которое должно содержать требования к разработке проекта, его цель, задачи, информацию об исполнителях и заказчиках, сроки и назначении, включает в себя описание предметной области проекта, основные определения и требования к результатам работы, порядку их сдачи и приёмки;
* руководство пользователя, которое должно содержать описание функций программы;
* отчет по УП, содержащий описание результатов выполненных работ в процессе разработки;
* программа и методика испытаний, которая предназначена для проверки выполнения функций программы, а также проверки соответствия требованиям технического задания.

Все вышеперечисленные документы должны быть написаны, согласно следующим требованиям:

* СТП ВятГУ 101-2004;
* ГОСТ 34.602−2020;
* ГОСТ 19 ЕСПД.

# Требования к приемо-сдаточным процедурам

Процесс приемки-сдачи в соответствии с установленными процедурами предусматривает следующий формат:

* Разработчик, являющийся автором проекта, прибывает в специально отведенную аудиторию, где представляет свою работу комиссии;
* Перед началом доклада разработчик предоставляет документацию, заверенную необходимым набором подписей, для ознакомления членам комиссии;
* Разработчик представляет проект в течение пяти минут;
* В рамках данного формата приемки проекта разработчик должен продемонстрировать полное понимание основных аспектов своей работы и быть готовым к обсуждению любых вопросов, которые могут возникнуть у членов комиссии;
* В результате процесса приемки-сдачи комиссия должна оценить выполненную работу и принять решение о её приемке/не приемке с занесением результатов в ведомость.
* Комиссия, ответственная за приемку работы, состоит из представителей заказчика:
* Чистяков Геннадий Андреевич – Руководитель образовательной программы по специальности 09.02.07 «Информационные системы и программирование».
* Ржаникова Елена Дмитриевна – Преподаватель по МДК 05.05 «Анализ и разработка технических заданий».
* Крутиков Александр Константинович – Преподаватель по учебной практике УП 05.
* Самоделкин Павел Андреевич – Преподаватель по МДК 06.01 «Внедрение информационных систем».