**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ**

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Департамент программной инженерии

|  |  |  |
| --- | --- | --- |
| СОГЛАСОВАНО  Приглашённый преподаватель департамента программной инженерии  факультета компьютерных наук НИУ ВШЭ  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.С. Смолин  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г. |  | УТВЕРЖДАЮ  Академический руководитель образовательной программы «Программная инженерия»  старший преподаватель департамента программной инженерии  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Н.А. Павлочев  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г. |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | ***Подп. и дата*** |  | | ***Инв. № дубл.*** |  | | ***Взам. инв. №*** |  | | ***Подп. и дата*** |  | | ***Инв. № подл*** |  | | **Рыболовный 2D симулятор на Unity –** **“Fishing in Belarus”**  **Техническое задание**  **ЛИСТ УТВЕРЖДЕНИЯ**  **RU.17701729.04.01-01 ТЗ 01-1-ЛУ** | | |
|  |  | |
| Исполнитель  Студент группы БПИ229  \_\_\_\_\_\_\_\_\_\_\_/Павлючик А.М./  \_\_\_\_\_\_\_\_\_\_\_/Степанова А. О./  «\_\_\_» \_\_\_\_\_\_\_\_\_\_2024 г. | |
|  | | |
|  | |  |

**Москва 2024**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| УТВЕРЖДЕН  RU.17701729.04.01-01 ТЗ 01-1-ЛУ | |  |  | |
| |  |  | | --- | --- | | ***Подп. и дата*** |  | | ***Инв. № дубл.*** |  | | ***Взам. инв. №*** |  | | ***Подп. и дата*** |  | | ***Инв. № подл*** |  | | **Рыболовный 2D симулятор на Unity –** **“Fishing in Belarus”**  **Техническое задание**  **RU.17701729.04.01-01 ТЗ 01-1**  **Листов** | | | | |
|  |  | | | |
|  | | | |
|  | | | | |
|  | | | |  |

**Москва 2024**

|  |  |
| --- | --- |
| ***Подп. и дата*** |  |
| ***Инв. № дубл.*** |  |
| ***Взам. инв. №*** |  |
| ***Подп. и дата*** |  |
| ***Инв. № подл*** |  |

СОДЕРЖАНИЕ

[**1. Введение** 5](#_Toc156823814)

[**1.1. Наименование программы** 5](#_Toc156823815)

[**1.2. Краткая характеристика области применения.** 5](#_Toc156823816)

[**2. Основания для разработки** 6](#_Toc156823817)

[**2.1. Документы, на основании которых ведется разработка** 6](#_Toc156823818)

[**2.2. Наименование темы разработки** 6](#_Toc156823819)

[**3. Назначение разработки** 7](#_Toc156823820)

[**3.1. Функциональное назначение** 7](#_Toc156823821)

[**3.2. Эксплуатационное назначение** 7](#_Toc156823822)

[**4. Требования к программе** 8](#_Toc156823823)

[**4.1. Требования к функциональным характеристикам** 8](#_Toc156823824)

[**4.1.1. Требования к составу выполняемых функций** 8](#_Toc156823825)

[**4.1.2. Требования к организации входных данных** 8](#_Toc156823826)

[**4.1.3. Требования к организации выходных данных** 8](#_Toc156823827)

[**4.2. Требования к интерфейсу** 8](#_Toc156823828)

[**4.3. Требование к надежности** 8](#_Toc156823829)

[**4.5. Условия эксплуатации** 8](#_Toc156823830)

[**4.6. Требования к составу и параметрам технических средств** 8](#_Toc156823831)

[**4.7. Требования к информационной и программной совместимости** 8](#_Toc156823832)

[**4.7.1. Требования к исходному коду** 9](#_Toc156823833)

[**4.8. Требования к маркировке и упаковке** 8](#_Toc156823834)

[**5. Требования к программной документации** 9](#_Toc156823835)

[**5.1. Предварительный состав программной документации** 9](#_Toc156823836)

[**5.2. Специальные требования к программной документации** 9](#_Toc156823837)

[**6. Технико-экономические показатели** 10](#_Toc156823838)

[**6.1. Ориентировочная экономическая эффективность** 10](#_Toc156823839)

[**6.2. Предполагаемая потребность** 10](#_Toc156823840)

[**6.3. Экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами** 10](#_Toc156823841)

[**7. Стадии и этапы разработки** 11](#_Toc156823842)

[**8. Порядок контроля и приёмки** 12](#_Toc156823843)

**Аннотация**

Техническое задание – это основной документ, оговаривающий набор требований и порядок создания программного продукта, в соответствии с которым производится разработка программы, ее тестирование и приемка.

Настоящее Техническое задание на разработку ПО «Рыболовный 2d симулятор на Unity – “Fishing in Belarus”» содержит следующие разделы: «Введение», «Основание для разработки», «Назначение разработки», «Требования к программе», «Требования к программной документации», «Технико-экономические показатели», «Стадии и этапы разработки», «Порядок контроля и приемки» и приложения.

В разделе «Введение» указано наименование и краткая характеристика области применения разработки.

В разделе «Основания для разработки» указан документ на основании, которого ведется разработка и наименование темы разработки.

В разделе «Назначение разработки» указано функциональное и эксплуатационное назначение программного продукта.

Раздел «Требования к программe» содержит основные требования к функциональным характеристикам, к надежности, к условиям эксплуатации, к составу и параметрам технических средств, к информационной и программной совместимости, к маркировке и упаковке, к транспортировке и хранению, а также специальные требования.

Раздел «Требования к программной документации» содержит предварительный состав программной документации и специальные требования к ней.

Раздел «Технико-экономические показатели» содержит ориентировочную экономическую эффективность, предполагаемую годовую потребность, экономические преимущества разработки

Раздел «Стадии и этапы разработки» содержит стадии разработки, этапы и содержание работ. В разделе «Порядок контроля и приемки» указаны общие требования к приемке работы. Настоящий документ разработан в соответствии с требованиями:

* ГОСТ 19.101-77 Виды программ и программных документов;
* ГОСТ 19.102-77 Стадии разработки;
* ГОСТ 19.103-77 Обозначения программ и программных документов;
* ГОСТ 19.104-78 Основные надписи;
* ГОСТ 19.105-78 Общие требования к программным документам;
* ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом;
* ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению.

Изменения к данному Техническому заданию оформляются согласно ГОСТ 19.603-78, ГОСТ 19.604-78.

Перед прочтением данного документа рекомендуется ознакомиться с терминологией, приведенной в Приложении 2 настоящего технического задания.

# 1. Введение

**1.1. Наименование программы**

Полное наименование программы – «Рыболовный 2d симулятор на Unity “Fishing in Belarus”».

Полное наименование программы на английском языке – «Fishing 2d Simulator on Unity “Fishing in Belarus”».

В документе будет использоваться сокращённая версия – «Fishing in Belarus»

**1.2. Краткая характеристика области применения.**

«Fishing in Belarus» – игровой проект на Unity, предназначенный для развлечения пользователя, путём предоставления возможности получения игрового опыта, симулирующего рыболовный процесс с национальными особенностями Республики Беларусь.

**2. Основания для разработки**

**2.1. Документы, на основании которых ведется разработка**

Документами, на основании которых ведётся разработка, являются: учебный план по подготовке специалистов на уровне бакалавриата по направлению 09.03.04 «Программная инженерия» и утверждённой академическим руководителем темы курсового проекта.

**2.2. Наименование темы разработки**

Наименование темы разработки – «Рыболовный 2d симулятора на Unity – “Fishing in Belarus”».

Наименование темы разработки на английском языке – «Fishing 2d Simulator on Unity – “Fishing in Belarus”».

Условное обозначение темы разработки – «Fishing in Belarus».

Программа выполняется в рамках темы курсовой работы «Рыболовный 2d симулятор на Unity – “Fishing in Belarus”» в соответствии с учебным планом подготовки специалистов бакалавриата по направлению 09.03.04 «Программная инженерия» Национального исследовательского университета «Высшая школа экономики», факультет компьютерных наук, департамент программной инженерии.

**3. Назначение разработки**

**3.1. Функциональное назначение**

«Fishing in Belarus» предоставляет пользователю возможность погрузится в виртуальный мир рыбалки в Беларуси. Игрок сможет выбирать локации, на которых рыбачить, а также зарабатывать деньги путём продажи пойманной рыбы. Деньги можно потратить на улучшение рыболовного инвентаря, еду и путешествия на другие локации для рыбалки.

**3.2. Эксплуатационное назначение**

Основным сценарием использования «Fishing in Belarus» является взаимодействие пользователя с программой с целью получения игрового опыта, основанного на симуляции рыбной ловли, для организации своего досуга.

Предполагаемой целевой аудиторией проекта являются геймеры старшего поколения, любители рыбалки и просто консервативные игроки.

**4. Требования к программе**

**4.1. Требования к функциональным характеристикам**

**4.1.1. Требования к составу выполняемых функций**

Приложение представляет собой единую программу, с которой пользователь и взаимодействует. Внутри игры можно провести условное разделение функционала на набор систем и механик, которые взаимодействуют между собой. Тогда весь функционал представим набором отдельных систем, описание которых представлено ниже.

*Механика голода игрока*

У игрока есть уровень сытости, при достижении нулевого значения которого игрок не может закидывать удочки, но при этом может вытянуть те, которые закинуты.

Пользователь может открыть инвентарь еды, в котором можно употребить доступные продукты, что увеличивает уровень сытости на количество единиц, соответствующее съеденному продукту.

Игрок может приобрести продукты в продуктовом магазине, который доступен на базе. Т.е. при нажатии на кнопку продуктового магазина появляется меню, в котором можно приобрести продукты, если у игрока имеется достаточное количество денег для этого.

*Механика игрового времени*

Игровое время представляет собой значение дня недели и текущего времени с информацией о часах и минутах. Обновление происходит спустя фиксированное время, т.е. игровое время дискретно и меняется на 5 минут игрового времени. В игре есть фазы дня и ночи. Ночью считается время с 22:00 до 6:00, днём, соответственно, - с 6:00 до 22:00.

*Механика прикормки*

Игрок, при достаточном количестве денег, может приобрести в магазине оснастки прикормку для ловли. Приобретённую прикормку можно посмотреть в меню прикормки. Выбрав удочку, которая заброшена, игрок может применить прикормку. Эффект от действия прикормки описан в пункте \*\*\* По истечению доступных порций в прикормке, она пропадает из меню прикормки.

*Механика снастей*

Игрок, при наличии достаточного количества денег, может приобрести следующие типы оснастки в магазине:

1. Удочка
2. Катушка
3. Леска
4. Крючок
5. Наживка
6. Прикормка

Все типы оснастки (кроме прикормки, она в отдельном меню, подробнее в пункте \*\*\*) игрок может посмотреть в меню оснастки. В этом же меню игрок может собрать из доступной оснастки готовое к забросу удилище.

Собранным удилищем считается набор из удочки, катушки, лески и крючка. В случае отсутствия одного и более элементов удилище считается несобранным и его нельзя достать для заброса.

Удочка, катушка и леска имеют уровень износа, он повышается при активном использовании удилища. При достижении максимального уровня износа снасть считается непригодной для ловли и её нельзя использовать для сборки удилища.

В случае, если одна или несколько частей собранного удилища максимально износились в момент, когда удилище достали, оно становится недоступным для заброса и единственная опция для него – убрать в инвентарь.

Если поломка какого-либо элемента происходит во время ужения рыбы, происходит следующее, в зависимости от сломанного элемента:

1. В случае поломки катушки во время ужения рыбы игрок теряет возможность использовать катушку для вытягивания.
2. В случае поломки удилища рыба срывается сразу.
3. В случае обрыва лески рыба срывается сразу, так же игрок теряет крючок.

Поломанная снасть отображается в инвентаре с пометкой о том, что она сломана. Игрок может её выкинуть или продать за небольшую цену прямо в меню оснастки.

*Механика садка и продажи рыбы*

У игрока есть меню садка, в котором показана вся рыба, которую игрок словил и не успел продать или выпустить. В меню можно посмотреть более подробную информацию о рыбе. Также её можно выпустить, засушить или сделать живца. Засушить можно рыбу массой от 100 до 300 грамм. Сделать живца можно из рыбы массой \*\*\*

*Механика перемещения по локациям*

Зайдя в меню путешествий на базе, игрок может выбрать локацию в Беларуси для переезда, если у него достаточный для этой локации разряд и хватает денег на путёвку.

В рамках одной локации игрок может бесплатно перемещаться по разным местам. Перемещение должно длиться соответствующее количество игрового времени, которое затрачивается на то, чтобы доехать до локации.

*Механика прогрессии статистики игрока* (меню игрока и обновление их данных)

У игрока есть меню рыболова, в котором находится следующая актуальная информация:

1. Разряд игрока
2. Никнейм
3. Любимая рыба
4. Любимая локация
5. \*\*\* добавить всякого

*Механика ловли рыбы* (закидывание, анимации)

После того, как игрок достал удочку, он ёе может з

*Локализация на 2 языка*

В меню настроек игрок может в любое время выбрать один из двух доступных языков:

1. Белорусский
2. Русский

Локализованы должны быть текст и текстуры.

*Аудио сопровождение*

Звуковое сопровождение в игре

*Механика заброса*

Каждое место локации имеет свою уникальную сетку глубины размера не менее 8 на 15. Вблизи берега глубина небольшая, чем дальше от берега, тем она больше. Достав удочку, игрок может закинуть её в любое место водоёма (сетки глубины), просто кликнув в какое-то место. В ячейках сетки глубины могут находится коряги и водоросли. Если заброс происходит в ячейку с корягой, то спустя некоторое время может произойти зацеп. Чтобы освободится от зацепа, игроку нужно произвести подсечку (ссылка на механику ужения), в момент подсечки с 50% шанцем может оборваться леска. Если заброс происходит в ячейку с водорослями, то происходит поклёвка (ссылка на механику клёва), но при этом нет сопротивления (ссылка на механику клёва) при ужении. Игрок может настроить глубину, на которую настроена удочка. Если глубина в том месте, куда игрок закинул поплавок, меньше, чем установленная, то поплавок принимает горизонтальное положение.

Механика *ужения*

На месте заброса появляется маркер в виде поплавка, игрок может подтянуть удочкой, крутить катушкой и сделать подсечку. При удержании кнопок подтягивания удочки и кручения катушки их напряжение растёт (скорость роста напряжения рассчитывается согласно параметрам рыбы). У игрока есть индикаторы нагрузки на удочку и леску, если они достигают максимального уровня – сразу выходят из строя.

Подсечка не влияет на уровень нагрузки удочки.

При кручении катушки маркер приближается к берегу до определённого момента (рассчитывается согласно сопротивлению рыбы). Чтобы появилась возможность тянуть дальше катушкой, после того как лимит приближения маркера исчерпан, нужно произвести подтягивание удочкой.

Основной геймплей заключается в ужении рыбы, балансировке между напряжениями лески и удочки, правильного подбора моментов для подтягивания и реакции на поведение рыбы.

*Механика симуляции поведения рыбы*

Каждое место локации содержит определённый набор видов рыб с параметрами, которые определяют то, какая рыба водится и на что клюёт. Если рыба обитает на этом месте, то имеют место следующие параметры:

1. Диапазон возможной массы рыбы
2. Диапазон активности на протяжении дня
3. Приоритетность наживок
4. Преобладающая глубина обитания с возможными допусками

Все эти параметры должны учитываться для расчёта времени поклёвки, а также параметры экземпляра рыбы, которая клюёт.

Поведение одного вида рыбы, но разных размеров должно отличаться. В качестве справочника по поведению рыбы можно использовать источники из Приложения \*\*\*

*Механика поклёвки*

После того, как игрок забросил удочку, на водоёме появляется маркер в виде поплавка, который не подвижен. Когда происходит поклёвка, поплавок начинает подтапливаться и всплывать, спустя какое-то время, которое определяется тем, насколько рыба агрессивна, поплавок полностью тонет. В этот момент игрок должен произвести подсечку, чтобы посадить рыбу на крючок и дальше пытаться её вытянуть. Если этого сделано не было, то поплавок приобретает горизонтальное положение, а на крючке не остаётся наживки.

*Механика рыболовной базы*

Игрок глобально может быть в двух состояниях, либо на рыболовном месте, где можно доставать удочки, либо на базе. База представляет собой место (панель), где игрок может перейти в нужные ему разделы, а именно:

1. Магазин снастей (тут ссылка на обьяснение выше)
2. Магазин еды (тут ссылка на обьяснение выше)
3. Место продажи рыбы (тут ссылка на обьяснение выше)
4. Агентство путешествий (тут ссылка на обьяснение выше)

*Меню настроек*

В любой момент игрового процесса пользователь может вызвать меню настроек, в котором можно:

1. Изменить язык приложения
2. Включить/выключить звук
3. Изменить уровень звука
4. Настроить управление

Так же меню настроек должно выполнять функцию паузы, т.е. игровое время и все процессы внутри игры останавливаются.

**4.1.2. Требования к организации входных данных**

Ввод данных в программу осуществляется с помощью пользовательского интерфейса игры и реализован посредством стандартных средств ввода: клавиатуры и мыши. Ввиду специфики игровых проектов, понятие «неверные входные данные» в контексте игры отсутствует. Есть набор сценариев, которые обрабатывают соответствующий ввод игрока. Если ввод не соответствует ни одному из сценариев игры, то он просто игнорируется.

**4.1.3. Требования к организации выходных данных**

Вывод данных производится на монитор в рамках пользовательского интерфейса программы. Других требований к организации выходных данных не выдвигается.

**4.2. Требования к интерфейсу**

Взаимодействие API.

**4.3. Требование к надежности**

В случае ошибок, возникающих при работе программы, игра должна уведомить пользователя о том, что произошёл сбой и предложить перезапустить игру.

Ситуации, при которых игра перестаёт обрабатывать ввод пользователя, зависает недопустимы.

**4.4. Требования к временным характеристикам**

Особых условий к временным характеристикам проекта не выдвигается.

**4.5. Условия эксплуатации**

Эксплуатация программы не должна требовать каких-то специальных навыков от пользователя. Достаточно владеть базовыми навыками работы с компьютером. Всю необходимую информацию по игре пользователь должен получать по ходу использования программы.

**4.6. Требования к составу и параметрам технических средств**

«Fishing in Belarus» для корректной работы требует компьютер с минимальными характеристиками, не превышающие следующие:

* Процессор: Intel Core i3-2100T либо аналог от AMD
* Объём ОЗУ: 4 Gb
* Объём ПЗУ: 5 Gb
* Видеоадаптер: GeForce GTX 550 Ti либо аналог от AMD

Программа должна быть работоспособной на устройствах под управлением ОС Windows любой версии, выше Windows 7.

**4.7. Требования к информационной и программной совместимости**

Игра реализуется на игровом движке Unity. Иных требований к информационной и программной совместимости не выдвигается.

**4.8. Требования к маркировке и упаковке**

Готовая программа поставляется в электронном формате: .exe файл для запуска и набора других файлов, необходимых для этого (при необходимости)

**4.9. Требования к транспортированию и хранению**

Программный пакет, заархивированный в формате .zip не должен превышать 2 Gb.

Иных требований к транспортированию и хранению не выдвигается.

**5. Требования к программной документации**

**5.1. Предварительный состав программной документации**

* «Рыболовный 2D симулятор – “Fishing in Belarus”».

Техническое задание (ГОСТ 19.201-78);

* «Рыболовный 2D симулятор – “Fishing in Belarus”».

Текст программы (ГОСТ 19.401-78);

* «Рыболовный 2D симулятор – “Fishing in Belarus”».

Пояснительная записка (ГОСТ 19.404-79);

* «Рыболовный 2D симулятор – “Fishing in Belarus”».

Программа и методика испытаний (ГОСТ 19.301-78);

* «Рыболовный 2D симулятор – “Fishing in Belarus”».

Руководство оператора (ГОСТ 19.505-79);

**5.2. Специальные требования к программной документации**

* Вся необходимая документация к проекту разрабатывается согласно ГОСТ 19.106-78 и ГОСТ к соответствующему документу (см. п. 5.1.).
* Пояснительная записка должна быть загружена в систему Антиплагиат «НИУ ВШЭ».
* Техническое задание и пояснительная записка, титульные листы остальных документов должны быть подписаны руководителем разработки и исполнителем
* Вся документация и проект сдается в электронном виде в формате .pdf или .docx. в архиве формата .rar или .zip.
* За три дня до защиты курсового проекта все следующие материалы:

1. Программный проект
2. Программная документация
3. Исполняемые файлы
4. Отзыв руководителя
5. Отчёт системы Антиплагиат

должны быть загружены через систему SmartLMS в дисциплину «Курсовой проект» одниv или несколькими архивами формата .rar или .zip.

# 6. Технико-экономические показатели

## **6.1. Ориентировочная экономическая эффективность**

В рамках данной работы расчет экономической эффективности не предусмотрен.

## **6.2. Предполагаемая потребность**

Пропуск дозы, неправильный тайминг приема, преждевременное прекращение приема, использование испорченных, просроченных лекарств – все это относится к нарушению режима медикаментозной терапии, которое может привести к различного рода последствиям, основные из которых - неэффективность лечения, ухудшение состояния здоровья и непредвиденные расходы на новые препараты для дополнительного лечения. В настоящее время, в развитых странах лишь в половине случаев прием курса лекарственных препаратов происходит без существенных нарушений. В развивающихся же странах ситуация усугубляется еще и недостаточным доступом населения к медицинской помощи. Один из возможных способов решения проблемы несоблюдения режима приема лекарственных препаратов – использование мобильных приложений и онлайн-сервисов для напоминаний и мониторинга домашней аптечки, которые могут помочь своим пользователям вовремя принимать лекарства и предоставлять им актуальную информацию о наличии необходимых препаратов дома. На рынке довольно мало решений, разработанных для Российской Федерации, которые отвечают требованиям и потребностям пользователей данного региона.

## **6.3. Экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами**

Анализ рынка показал, что решений, которые ставят управление электронной домашней аптечкой в качестве первичного функционала приложения – нет. Такой функционал если и возникает, то как вторичный продукт в приложениях группы “Pills Reminders”, которые ставят на первое место функцию напоминания о времени приема препарата по заранее заданному расписанию. В данном решении на первое место как раз-таки выдвигается функционал электронной домашней аптечки.

Решение должно поддерживать платформу ОС Android, должно быть локализовано для использования на территории Российской Федерации и распространяться бесплатно.

**7.** **Стадии и этапы разработки**

|  |  |  |  |
| --- | --- | --- | --- |
| **Стадии разработки** | **Этапы разработки** | **Содержание работ** | **Срок выполнения** |
| 1. Техническое задание | Обоснование необходимости разработки проекта | Постановка задачи | 01.09.2023 – 01.10.2023 |
| Сбор необходимой информации | 01.10.2023 – 01.10.2023 |
| Обоснование эффективности потенциального решения | 01.10.2023 – 13.11.2023 |
| Разработка и утверждение технического задания | Выработка требований к проекту | 14.11.2023 –  01.01.2024 |
| Определение этапов разработки программы и программной документации | 02.01.2024 –  15.01.2024 |
| Согласование и утверждение технического задания | 16.01.2024 –  15.02.2024 |
| 2. Рабочий проект | Разработка программы | Проектирование архитектуры приложения | 16.02.2024 – 01.03.2024 |
| Написание кода программы | 02.03.2024 – 15.03.2024 |
| Отладка проекта | 16.03.2024 –  01.04.2024 |
| Разработка программной документации | Написание программной документации к проекту в соответствии с требованиями | 02.04.2024 –  10.04.2024 |
| Испытания программы | Выработка и согласование программы и методики испытаний | 11.04.2024 –  18.04.2024 |
| Проведение испытаний | 19.04.2024 –  24.04.2024 |
| Изменение программы и программной документации согласно результатам испытаний | 25.04.2024 –  01.05.2024 |
| 3. Внедрение | Подготовка и передача результатов работы | Подготовка материалов к передаче | 01.05.2024 – 10.05.2024 |
| Передача материалов разработки | 10.05.2024 –  30.05.2024 |

**8. Порядок контроля и приёмки**

**8.1 Виды испытаний**

Целью испытаний является выявления отклонений программы от функциональных требований технического задания. Соответственно, испытания программы включают в себя следующие модули:

1. Блок 1
2. Блок 2
3. Блок 3
4. Блок 4

**8.2 Общие требование к приёмке работы**

Контроль и приемка программного продукта осуществляются в соответствии с документом Программа и методика испытаний (ГОСТ 19.301-79). Прием программы будет утвержден при корректной работе программы в соответствии с пунктом 4.1.1 при различных входных данных, соответствующих условиям в пункте 4.1.2 данного документа и при предоставлении полной документации к продукту, указанной в пункте 5.1, выполненной в соответствии с требованиями, указанными в пункте 5.2 данного технического задания.

# ПРИЛОЖЕНИЕ 1

**СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ**

1. ГОСТ 19.101-77: Виды программ и программных документов. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
2. ГОСТ 19.102-77: Стадии разработки. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
3. ГОСТ 19.103-77: Обозначения программ и программных документов. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
4. ГОСТ 19.104-78: Основные надписи. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
5. ГОСТ 19.105-78: Общие требования к программным документам. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
6. ГОСТ 19.106-78: Требования к программным документам, выполненным печатным способом. //

Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

1. ГОСТ 19.201-78: Техническое задание. Требования к содержанию и оформлению. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
2. ГОСТ 19.603-78: Общие правила внесения изменений. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
3. ГОСТ 19.604-78: Правила внесения изменений в программные документы, выполненные печатным способом. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

# ПРИЛОЖЕНИЕ 2

# ТЕРМИНОЛОГИЯ

|  |  |
| --- | --- |
| **Термин** | **Определение** |
| Игровой проект | прикладной программный продукт, предназначенный для развлечения пользователя, путём предоставления возможности игроку взаимодействия с виртуальным миром, существующим согласно игровым механикам. |
| Игровая механика |  |
| Игрок |  |
| Игровой опыт |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ** | | | | | | | | | |
| Номера листов (страниц) | | | | | Всего листов (страниц  ) в докум. | № документа | Входящий  №сопровод ительного докум. и дата | Подп. | Да- та |
| И  з м. | Изменён ных | Заменён ных | Новых | Аннули рованны х |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |