Игра «Питон» («Змейка»).

Техническое задание

15 листов

Оглавление

[ВВЕДЕНИЕ 3](#_Toc152668777)

[1.ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ 3](#_Toc152668778)

[2.НАЗНАЧЕНИЕ РАЗРАБОТКИ 3](#_Toc152668779)

[3. ТРЕБОВАНИЯ К ПРОГРАММЕ ИЛИ ПРОГРАММНОМУ ИЗДЕЛИЮ 4](#_Toc152668780)

[4. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ 6](#_Toc152668781)

[5. ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ 8](#_Toc152668782)

[6. СТАДИИ И ЭТАПЫ РАЗРАБОТКИ 10](#_Toc152668783)

[7. ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ 12](#_Toc152668784)

[8. СРОКИ ВЫПОЛНЕНИЯ 15](#_Toc152668785)

# ВВЕДЕНИЕ

Данное техническое задание разрабатывается в соответствии с требованиями ГОСТ 19.106-78 и представляет собой виртуальную игру «Питон» («Змейка»). Целью этого проекта является разработка компьютерной игры с использованием языка программирования Python и графической библиотеки Pygame. "Питон" предоставляет пользователю возможность не только насладиться ретро-геймплеем, но и испытать свои навыки в управлении змейкой, увеличивая ее длину и избегая столкновений со стенами и самим собой.

1.ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ.

Основания для разработки данной программы закладываются в неотъемлемой необходимости совершенствования и эффективного управления железнодорожными системами. Развитие транспортной инфраструктуры и повышение объемов перевозок требуют надежных инструментов для анализа и оптимизации работы сортировочных узлов, играющих ключевую роль в организации движения по железной дороге.

# 2.НАЗНАЧЕНИЕ РАЗРАБОТКИ

Целью данной разработки является создание компьютерной игры "Питон" (или "Змейка"), предназначенной для развлечения пользователей и предоставления им увлекательного геймплейа. Игра разрабатывается с использованием языка программирования Python и графической библиотеки Pygame с целью обеспечения простоты использования, кроссплатформенности и приятного визуального опыта. Главные задачи разработки включают в себя:

1. Реализация игрового процесса: Создание механики управления змейкой, ее роста при поедании объектов и генерации новых объектов на игровом поле.

2. Графический интерфейс: Разработка визуального оформления, включая отображение змейки, объектов, счетчика очков, а также обеспечение адаптации под различные разрешения экранов.

3. Управление и пользовательский опыт: Создание удобного управления змейкой с клавиатуры, добавление звукового сопровождения, а также внедрение меню с возможностью начать новую игру, выйти из игры и настроить параметры.

4. Техническая стабильность: Обеспечение безопасности и стабильной работы приложения путем тщательного тестирования на различных платформах.

5. Оптимизация и уровни сложности: Работа над оптимизацией кода для обеспечения плавного геймплея даже на слабых компьютерах, а также создание системы уровней, увеличивающей сложность игры по мере продвижения игрока.

# 3. ТРЕБОВАНИЯ К ПРОГРАММЕ ИЛИ ПРОГРАММНОМУ ИЗДЕЛИЮ

3.1. Функциональные требования:

3.1.1. Игровой процесс

3.1.1.1. Управление змейкой:

Игрок должен иметь возможность управлять движением змейки с использованием клавиш управления (вверх, вниз, влево, вправо).

3.1.1.2. Рост змейки. При контакте змейки с объектом, представляющим еду, змейка должна увеличивать свою длину.

3.1.1.3. Генерация объектов. На игровом поле периодически должны появляться объекты (еда), которые змейка может съесть.

3.1.2. Игровое поле

3.1.2.1. Размеры поля

Игровое поле должно иметь заданные размеры, обеспечивающие удобство игры.

3.1.2.2. Стены

Змейка и объекты не должны проникать сквозь стены игрового поля.

3.1.3. Графический интерфейс

3.1.3.1. Отображение змейки

Змейка должна быть ярко и наглядно отображена на игровом поле.

3.1.3.2. Отображение объектов

Объекты (еда) должны быть четко видны на игровом поле.

3.1.3.3. Счетчик очков

Игра должна включать в себя счетчик очков, отображающий текущее количество съеденных объектов.

3.1.4. Управление

3.1.4.1. Управление с клавиатуры

Управление змейкой должно быть реализовано с использованием стрелок на клавиатуре.

3.1.5. Система уровней

3.1.5.1. Увеличение сложности

С увеличением количества съеденных объектов уровень сложности игры должен постепенно увеличиваться.

3.1.6. Звуковое сопровождение

3.1.6.1. Звуковые эффекты

Игра должна включать в себя звуковое сопровождение, такое как звук при поедании объекта.

3.1.7. Меню

3.1.7.1. Основное меню

Игра должна иметь главное меню с опциями: "Начать новую игру", "Выйти из игры" и "Настройки".

3.1.7.2. Настройки

Меню настроек должно позволять пользователю изменять параметры, такие как уровень сложности или звуковые настройки.

Эти функциональные требования обеспечат создание полноценной игры "Питон", удовлетворяющей ожиданиям пользователя и предоставляющей приятный игровой опыт.

# 4. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

4.1. Техническая документация:

Общие сведения:

Полное и точное наименование программы с указанием версии и даты выпуска.

Краткое описание основных целей и возможностей программы.

Установка и настройка:

Подробная инструкция по установке программы.

Рекомендации по конфигурации программы для оптимальной работы.

4.2. Руководство пользователя:

Установка и запуск:

Описать процедуру установки игры "Питон".

Обеспечить инструкции по запуску приложения на различных операционных системах.

Основные элементы интерфейса:

Предоставить подробное описание элементов интерфейса, таких как змейка, объекты, счетчик очков, их функциональность и взаимодействие.

Управление:

Предоставить пользователю подробную информацию об управлении змейкой с клавиатуры.

Система уровней:

Объяснить, как система уровней влияет на сложность игры с увеличением съеденных объектов.

Настройки:

Информировать пользователя о возможности настройки параметров игры, таких как уровень сложности и звуковые настройки.

Меню:

Описать функциональность каждого пункта основного меню, а также дать рекомендации по их использованию.

4.1.2. Разработчику

Архитектура приложения:

Предоставить обзор структуры программы, классов и основных функций.

Технические детали:

Обеспечить подробное описание технических аспектов, таких как алгоритм движения змейки, генерация объектов, управление событиями.

API и библиотеки:

Указать использованные API и библиотеки, в том числе версии, с целью обеспечения легкости поддержки и разработки.

Руководство по тестированию:

Предоставить инструкции по тестированию различных компонентов игры, включая тестирование безопасности и производительности.

Оптимизация кода:

Дать рекомендации по оптимизации кода с целью обеспечения плавного геймплея и эффективного использования ресурсов.

Расширение функционала:

Описать возможности для расширения функционала игры, добавления новых элементов и улучшения существующих.

4.1.3. Техническая спецификация

Язык программирования:

Указать, что игра разработана на языке программирования Python.

Графическая библиотека:

Указать использование Pygame (или аналогичной библиотеки) для создания графического интерфейса.

Поддержка кроссплатформенности:

Обеспечить информацию о поддержке различных операционных систем (Windows, Linux, macOS).

Версионирование:

Указать используемую систему контроля версий (например, Git) и рекомендации по работе с ней.

Зависимости:

Предоставить список всех зависимостей, включая версии, необходимых для успешной компиляции и запуска игры.

Требования к аппаратному обеспечению:

Указать минимальные и рекомендуемые требования к аппаратному обеспечению для оптимальной работы игры.

# 5. ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

5.1. Затраты на разработку:

Затраты на трудовые ресурсы:

Оплата разработчиков, включая программистов, дизайнеров и тестировщиков.

Расчет производится на основе часовой ставки и планируемого времени выполнения проекта.

Затраты на программное обеспечение:

Покупка лицензий на необходимые программные средства разработки и графические инструменты.

Затраты на оборудование:

При необходимости, закупка компьютеров и другого оборудования для разработчиков.

Затраты на документацию:

Расходы на создание технической документации, руководства пользователя и других документов.

Затраты на тестирование:

Финансирование тестирования продукта на различных платформах и в различных условиях.

Затраты на маркетинг:

Рекламные расходы для продвижения игры после завершения разработки.

Накладные расходы:

Расходы на административные нужды, аренду офиса (если применимо), коммунальные услуги.

Резервный фонд:

Запасные средства на случай непредвиденных обстоятельств и возможных изменений в процессе разработки.

Прочие затраты:

Различные небольшие расходы, такие как покупка шрифтов, изображений и других ресурсов.

5.2. Прогнозируемая прибыль и экономический эффект:

Рыночный анализ:

Исследование аналогичных игр в жанре и оценка их коммерческого успеха.

Определение целевой аудитории и ее потенциала для игры "Питон".

Маркетинговые стратегии:

Разработка эффективной маркетинговой кампании для продвижения игры.

Привлечение внимания к игре через различные каналы, включая социальные сети, игровые платформы и рекламные компании.

Ценовая стратегия:

Определение оптимальной цены для игры, учитывая конкурентоспособность на рынке и ожидаемое значение для потребителей.

Продажи и дистрибуция:

Разработка стратегии продаж, включая выбор платформ для дистрибуции (например, Steam, App Store).

Партнерство с платформами для увеличения видимости и распространения игры.

Прогноз продаж:

Оценка прогнозируемых продаж на основе рыночного анализа и маркетинговых усилий.

Экономический эффект:

Оценка ожидаемого дохода от продаж и учет всех затрат на разработку и маркетинг.

Расчет экономического эффекта в виде чистой прибыли.

Оценка возврата инвестиций (ROI):

Оценка времени, необходимого для окупаемости затрат на разработку.

Стратегии монетизации:

Разработка стратегий монетизации, таких как дополнительные контенты, микро транзакции или подписки.

# 6. СТАДИИ И ЭТАПЫ РАЗРАБОТКИ

6.1. Определение требований и составление технического задания:

6.1.1. Стадия 1: Анализ требований

6.1.1.1. Этап 1.1: Сбор требований

Задача: провести встречи с заказчиком и заинтересованными сторонами для определения основных требований к игре "Питон".

Действия:

Провести интервью с заказчиком для выяснения функциональных и нефункциональных требований.

Исследовать рынок аналогичных игр и выявить успешные решения.

6.1.1.2. Этап 1.2: Анализ конкурентов

Задача: проанализировать конкурентов в жанре аркадных игр и извлечь уроки для улучшения проекта.

Действия:

Изучить популярные аркадные игры, особенно те, которые используют подобный геймплей.

Определить их преимущества и недостатки.

6.1.2. Стадия 2: Составление технического задания

6.1.2.1. Этап 2.1: Определение функциональных требований

Задача: определить основные функции и возможности игры.

Действия:

Составить список функциональных требований, включая управление, игровой процесс, графический интерфейс и дополнительные возможности.

6.1.2.2. Этап 2.2: Уточнение технических деталей

Задача: уточнить технические детали, такие как выбор языка программирования и графической библиотеки.

Действия:

Провести исследование наилучших практик для разработки аркадных игр.

Принять решение о выборе языка программирования и библиотеки.

6.1.2.3. Этап 2.3: Создание структуры данных

Задача: разработать структуру данных для хранения информации о змейке, объектах и других элементах игры.

Действия:

Создать схему данных, определить классы и методы для основных игровых объектов.

6.1.2.4. Этап 2.4: Разработка архитектуры

Задача: разработать общую архитектуру приложения.

Действия:

Определить основные компоненты, связи между ними и общий поток управления в игре.

6.1.2.5. Этап 2.5: Написание технического задания

Задача: сформулировать техническое задание с учетом всех вышеописанных требований и деталей.

Действия:

Составить подробное техническое задание с указанием всех функциональных и технических требований.

6.1.3. Стадия 3: Планирование разработки

6.1.3.1. Этап 3.1: Разработка графика работы

Задача: составить график работы, определить сроки и этапы разработки.

Действия:

Разбить процесс разработки на этапы и назначить сроки каждому этапу.

Учесть возможные риски и предусмотреть резервное время.

6.1.3.2. Этап 3.2: Распределение задач

Задача: распределить задачи между участниками команды разработки.

Действия:

Определить ответственных за каждый функциональный блок игры.

Убедиться, что каждый разработчик понимает свои обязанности.

6.1.3.3. Этап 3.3: Определение тестовых сценариев

Задача: определить сценарии тестирования для проверки функциональности и производительности игры.

Действия:

Создать сценарии для тестирования различных аспектов игры, включая управление, графику и взаимодействие объектов.

# 7. ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ

7.1. Внутренний контроль

7.1.1. Этап 1: Разработка

7.1.1.1. Задача 1.1: Написание кода

Контрольные мероприятия:

Разработчик должен придерживаться установленных стандартов кодирования.

Использование комментариев для пояснения сложных участков кода.

Валидация:

Проведение регулярных код-ревью между членами команды.

7.1.1.2. Задача 1.2: Тестирование

Контрольные мероприятия:

Разработчик должен провести модульное тестирование своего кода перед интеграцией в общий репозиторий.

Валидация:

Запуск автоматизированных тестов для проверки корректности функционала.

7.1.2. Этап 2: Интеграция

7.1.2.1. Задача 2.1: Объединение кода

Контрольные мероприятия:

Проведение интеграционного тестирования после объединения кода различных разработчиков.

Валидация:

Проверка взаимодействия различных компонентов игры.

7.1.2.2. Задача 2.2: Тестирование совместимости

Контрольные мероприятия:

Проверка совместимости игры на различных операционных системах.

Валидация:

Тестирование на Windows, Linux и macOS.

7.1.3. Этап 3: Тестирование

7.1.3.1. Задача 3.1: Тестирование безопасности

Контрольные мероприятия:

Проведение тестов на выявление уязвимостей и защиту от возможных атак.

Валидация:

Проверка результатов тестов безопасности.

7.1.3.2. Задача 3.2: Тестирование производительности

Контрольные мероприятия:

Оценка производительности игры при различных нагрузках.

Валидация:

Сравнение результатов тестирования с установленными стандартами производительности.

7.1.4. Этап 4: Оптимизация

7.1.4.1. Задача 4.1: Улучшение производительности

Контрольные мероприятия:

Использование профилировщиков для выявления узких мест в производительности.

Валидация:

Улучшение производительности в соответствии с выявленными проблемами.

7.1.4.2. Задача 4.2: Оптимизация кода

Контрольные мероприятия:

Проведение код-ревью с акцентом на оптимизацию.

Валидация:

Проверка изменений в производительности после оптимизации.

7.1.5. Этап 5: Документирование

7.1.5.1. Задача 5.1: Написание технической документации

Контрольные мероприятия:

Постоянное обновление технической документации по мере разработки.

Валидация:

Проверка полноты и актуальности документации.

7.1.5.2. Задача 5.2: Разработка руководства пользователя

Контрольные мероприятия:

Создание понятного и информативного руководства пользователя.

Валидация:

Тестирование руководства на пользователях для проверки понятности и полезности.

# 8. СРОКИ ВЫПОЛНЕНИЯ

8.1. Определение требований и составление технического задания:

8.1.1. Этап 1: Анализ требований

8.1.1.1. Этап 1.1: Сбор требований

Сроки:

Начало: [01.12.2023].

Окончание: [27.12.2023].

8.1.1.2. Этап 1.2: Анализ конкурентов

Сроки:

Начало: [02.12.2023].

Окончание: [03.12.2023].

8.1.2. Этап 2: Составление технического задания

8.1.2.1. Этап 2.1: Определение функциональных требований

Сроки:

Начало: [03.12.2023].

Окончание: [05.12.2023].

8.1.2.2. Этап 2.2: Уточнение технических деталей

Сроки:

Начало: [05.12.2023].

Окончание: [07.12.2023].

8.1.2.3. Этап 2.3: Создание структуры данных

Сроки:

Начало: [08.12.2023].

Окончание: [12.12.2023].

8.1.2.4. Этап 2.4: Разработка архитектуры

Сроки:

Начало: [15.12.2023].

Окончание: [19.12.2023].

8.1.2.5. Этап 2.5: Написание технического задания

Сроки:

Начало: [22.12.2023].

Окончание: [27.12.2023].

Общие замечания:

Все даты должны быть согласованы с командой разработки и заказчиком.

Сроки выполнения каждого этапа могут зависеть от сложности проекта и доступности ресурсов.