Министерство образования Калининградской области

государственное бюджетное учреждение Калининградской области

профессиональная образовательная организация

«Колледж информационных технологий и строительства»

(ГБУ КО ПОО «КИТиС»)

**Отчет по учебной практике**

УП.04 Сопровождение и обслуживание программного обеспечения компьютерных систем

по ПМ.04 Сопровождение и обслуживание программного обеспечения компьютерных систем

Специальность 09.02.07 «Информационные системы и программирование»

Сроки прохождения практики:

с «13» Февраля 2023 г. по «6» Марта 2023 г.

Место практики ГБУ КО ПОО «КИТиС»

|  |  |
| --- | --- |
| Выполнил: | студент 2 курса,  группы ИСп 21-2К  Хорунов Иван Павлович  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |
| Проверила: | Большакова-Стрекалова Анна Викторовна  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (оценка)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) |

Калининград 2023

Содержание

1 Обеспечение внедрения и поддержки программного обеспечения компьютерных систем

1.1 Инструктаж по технике безопасности

1.2 Описание выбранного аппаратного и программного обеспечения

1.3 Методы внедрения и анализа функционирования программного обеспечения

1.4 Сценарий и техническое задание на внедрение ПО

1.5 Загрузка и установка программного обеспечения

2 Проблемы совместимости и методы их устранения

2.1 Тестирование на совместимость в безопасном режиме

2.2 Инструменты повышения производительности программного обеспечения

2.3 Средства диагностики оборудования

2.4 Измерение и анализ эксплуатационных характеристик качества программного обеспечения

2.5 Оптимизация и модификация ПО

3 Этапы сопровождения и обслуживания ПО

3.1 Разработка технического задания на сопровождение

3.2 Обеспечение качества компьютерных систем в процессе эксплуатации

3.3 Разработка сопровождающей документации

4 Методы и средства защиты компьютерных систем

4.1 Тестирование на ошибки ввода/вывода данных

4.2 Тестирование интерфейса

4.3 Тестирование контента

4.4 Анализ рисков. Выявление первичных и вторичных ошибок

4.5 Обнаружение вируса и устранение последствий его влияния

4.6 Тестирование защиты программного обеспечения

4.7 Работа с реестром. Работа с программой восстановления файлов и очистки дисков

5. Индивидуальное задание [9]

5.1 Практические работы

5.2 Порядок установки и сопровождения клиентского программного обеспечения

5.3 Связь сопровождения с эволюцией программного обеспечения

Введение

Конечная цель многих разработчиков игр и визуализации – создание красивых и увлекательных проектов, которые могут захватить воображение пользователей и создать неповторимый опыт. Для достижения этой цели, многие разработчики используют мощные инструменты, такие как Unreal Engine. Unreal Engine – это мощный игровой движок, созданный компанией Epic Games, который используется для разработки игр, виртуальной и дополненной реальности, а также визуализации архитектурных проектов, мультимедийных и киноиндустрий.

Внедрение Unreal Engine в проект может значительно упростить процесс разработки, улучшить производительность и повысить качество графики и анимации. Кроме того, Unreal Engine предоставляет широкий спектр инструментов и библиотек, которые позволяют разработчикам создавать сложные и увлекательные игры с высокой степенью интерактивности и реалистичности. Однако, внедрение Unreal Engine требует от разработчиков определенных знаний и навыков, включая знание языка программирования C++ и основных принципов разработки игр.

Если вы решили внедрить Unreal Engine в свой проект, важно продумать его архитектуру и принять решения о выборе наиболее подходящих компонентов и настроек. Начиная с создания прототипа и заканчивая финальной отладкой и оптимизацией, процесс внедрения Unreal Engine требует тщательного планирования и совместной работы команды разработчиков, чтобы достичь успеха.

1Обеспечение внедрения и поддержки программного обеспечения компьютерных систем

* 1. Инструктаж по технике безопасности

Первым делом хотелось бы рассказать о технике безопасности при использовании Unreal Engine. Прежде всего, не стоит скачивать данное ПО со сторонних сайтов, Unreal Engine скачивается с официального сайта [www.unrealengine.com](http://www.unrealengine.com/). Что будет описано в последующих пунктах. Лучше всего, перед скачиванием и установкой Unreal engine, Установить антивирусное ПО. Это поможет защитить ваш компьютер от вирусов, троянов и друх видов вредоносного ПО. После установки рекомендуется следить за обновлениями. Установка покетов обновлений поможет устранить уязвимости, которые могут быть использованы злоумышлинниками в сваих карыстных целях. Также не мало выажным, является соблюдение правил эксплуатации оборудования. Регулярное обслуживание и правильное использование оборудования помогут избежать его поломок и сбоев, что может привести к потере данных и другим проблемам.

Это основные правила техники безопасности, которые следует соблюдать при работе с Unreal Engine. Соблюдение выше описанных правил поможет вам защитить вас и ваши данные от угроз и обеспечить более безопасное использование ПО.

* 1. Описание выбранного аппаратного и программного обеспечения

Unreal Engine – игровой движок разрабатываемый и поддерживаемый компанией Epic Games. Этот движок один из популярных в мире. Наиболее известная версия этого движка это Unreal Engine 5. С каждым обнавлением графика становится всё более мощной, возможности Повышаются и позволяют создавать все более сложные и реалистичные игры. В основном Unreal Engine использьзуется в создании игр по ПК, консоли и мобильные устройства. Главная цель внедрения Unreal Engine заключается в том, чтобы минимизировать риски, а также понизить стоимость разработки программного продукта. К использованию рекомендуется новая (На данный момент) версия Unreal Engine 5.1.1 в ней есть множество новых функций которых небыло в прошлых версиях, также в этой версии, что не мало важно повысели производительность.

Перед созданием протатипа следует разработать логику игры прыжки, стрельбу и т.д. При разработке логики следует использовать Blueprint. Blueprint – Это визуальная, нодовая система програмирования, которая используется в Unreal Engine.

После всего выше перечисленого, следует сделать прототип прокта. После создания играбильного прототипа следует создавать контент для прототипа. Следует добовлять новые текстуры, модели, звуки и другие элементы. Когда контент для прототипа готов, то следует приступать к тестированию прототипа. Проверьте проект на соотвецтвие требованиям, а также следует исправить некоторые недочёты проекта. После всего выше перечисленного нужно развернуть проект на выбранной вами платформе.

Unreal Engine требовательный к ресурсам компьютера движок. Поэтому аппаратное обеспечение к ниму нужно соответствующее:

Процессор: AMD 8 -core Ryzen 7 5800X 3.8-4.7 ГГц

Видео карта: GeForce RTX 3070 8ГБ

Оперативная память: 16ГБ DDR4

Объём накопителя: 1ТБ SSD

Блок питания: 700W 80+ Bronze

* 1. Методы внедрения и анализа функционирования программного обеспечения

Unreal Engine как описывалось ранее – это мощный игровой движок, который используется для разработки игр и не только. Для внедрения Unreal Engine необходимо выполнить следующие основные шаги.

1. Регистрация – Перез скачиванием нужно зарегистрироватся на оффициальном сайте, чтобы зайти в движок под своим аккаунтом.

2) Загрузка и установка – Скачать Unreal Engine с оффициального сайта а затем установить

3) Выбрать версию ПО – После установки проекта, нужно выбрать любую доступную версию движка, а затем установить.

4) Настройка проекта – После выбора версии и её установки, нужно настроить прект а именно

4.1) Выбрать шаблон проекта

4.2) Выбрать BLUEPRINT при создании прокта – Если вам Blueprint не нужен, то тогда

4.3) Выбрать платформу на которой будет работать игра нужно будет выбрать C++ и докачать Visual Studio 2019

4.4) Выбрать Quality Preset – Лучше выбрать Scalable при создании проекта. Так как Maximum очень требователен к аппаратному обеспечению.

4.5) Пункт Стартовый контент – Этот пункт можно не включать. Он в свою очередь добавит в папку с проектом основные текстуры и материалы.

4.6) Пункт Raytracing – Эта настройка означает, будет ли в вашем проекте новая, но требовательная к аппаратному обеспечению технология процедурной генирации света. Эта функция работает не на всех видео картах.

4.7) Выбрать место на диске где будет хранится проект

4.8) Ввести имя проекта (Латиницей и без пробелов)

После всего выше описанного нужно создать проект с выбранными пораметрами.

* 1. Сценарий установки Unreal Engine

1) Зайти на оффициальный сайт Unreal Engine и скачать лаунчер Epic Games.

2) Установить лаунчер Epic Games, предворительно выбров место хранения на диске.

3) После устоновки следует войти в лаунчер под соим личным Google аккаунтом.

4) Зайтив раздел Unreal Engine, во вкладку библиотека.

5) Где написано «ВЕРСИИ UNREAL ENGINE» нажать на +

6) Выбрать версию для устанвоки и установить

Сроки: от 1 месяца до 1 года

Расходы: 10000$

Отвецтвенные лица: Вася Пупкин, Павел Павлов

1.5 Загрузка и установка программного обеспечения

1) Зайти на оффициальный сайт Unreal Engine, затем нажать на кнопку “Download Launcher”.

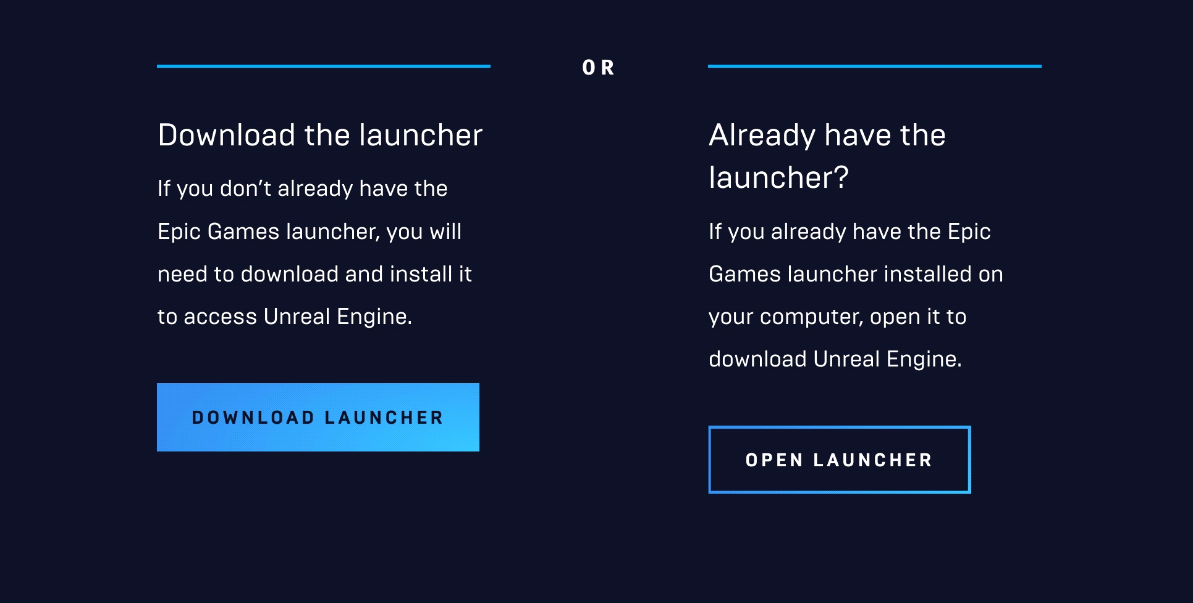


Рисунок 1 (Сайт [www.unrealengine.com](http://www.unrealengine.com/))

1. Запустить установщик и выбрать место на диске.

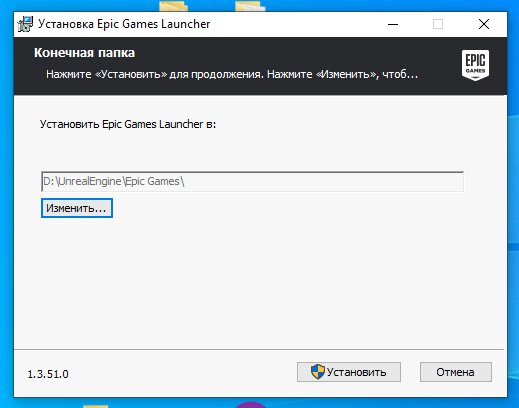


Рисунок 2 (Установщик)

1. Дождаться, окончания установки.

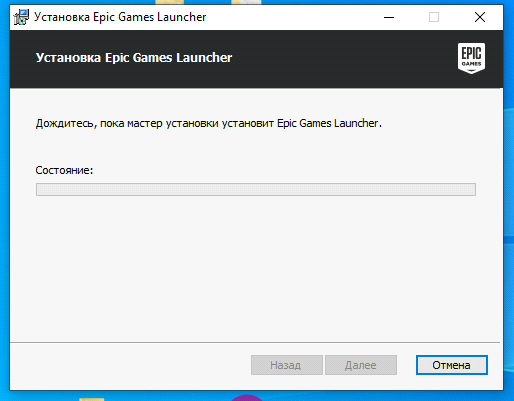


Рисунок 3 (Установщик)

4) Ожидаем окончания, загрузки обновлений.

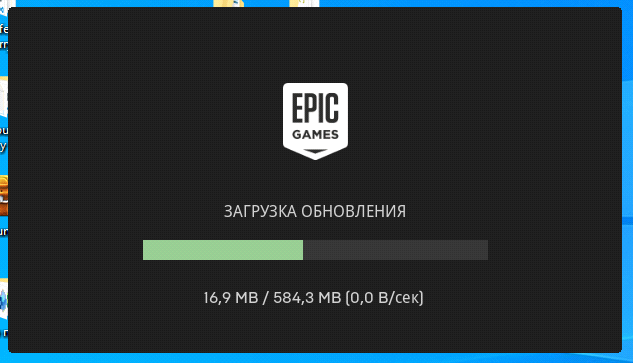


Рисунок 4 (Обновление Epic Games)

5) Выбрать аккаунт для входа.

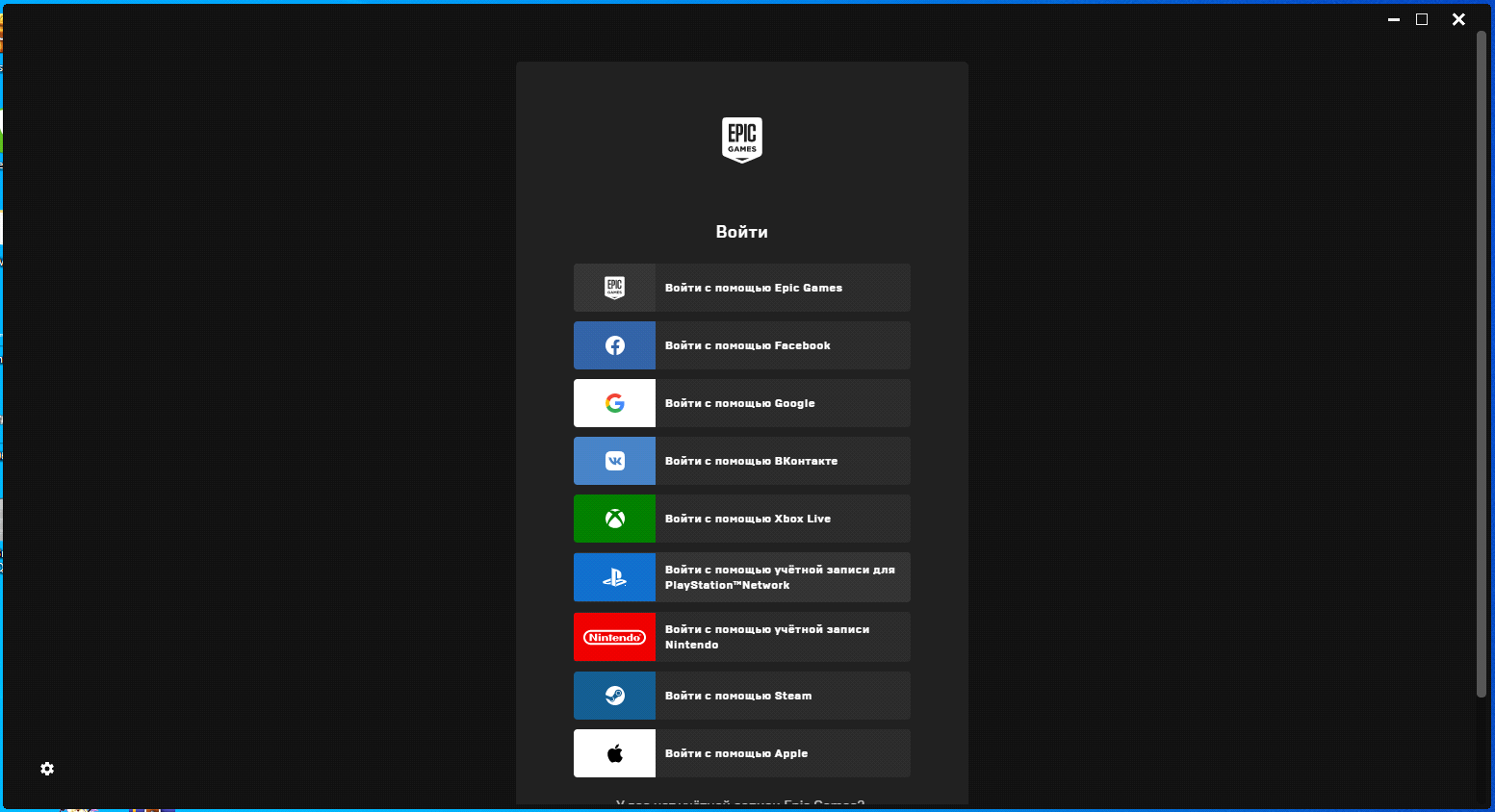


Рисунок 5 (Вход в аккаунт Epic Games)

6) Зайти в раздел “Unreal Engine” во вкладку “Библиотека”. Нажать на кнопку “+”, выбрать версию Unreal Engine и дождаться установки.

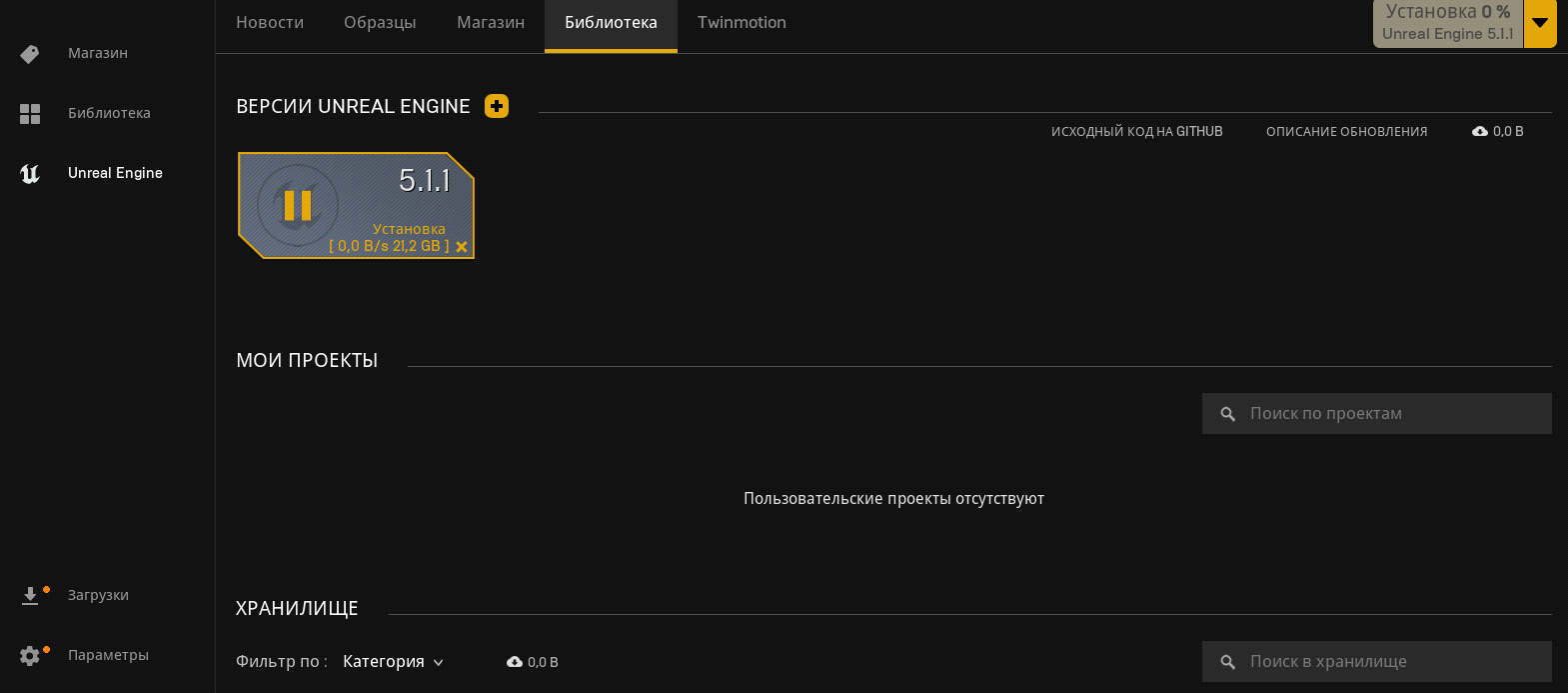


Рисунок 6 (Установка Unreal Engine)

* Проблемы совместимости и методы их устранения

2.1 Тестирование на совместимость в безопасном режиме

* Перейдите в меню "Пуск" и выберите "Выключить компьютер". Удерживайте клавишу Shift и нажмите кнопку "Перезагрузить", чтобы перейти в меню "Дополнительные параметры загрузки".
* Выберите "Отладка" и затем "Безопасный режим". Это запустит компьютер в безопасном режиме.
* Запустите Unreal Engine и убедитесь, что он работает корректно. Проверьте, что вы можете создавать новые проекты, открывать и редактировать существующие проекты, запускать проекты на выполнение, а также использовать все инструменты и функции, которые обычно доступны в Unreal Engine.
* Проверьте, что все плагины, материалы и другие ресурсы, которые вы используете в своих проектах, работают без ошибок и не вызывают сбоев в безопасном режиме.
* Если вы заметили какие-либо проблемы или ошибки в процессе тестирования, попробуйте устранить их.
* После завершения тестирования, выйдите из безопасного режима, перезагрузите компьютер и запустите Unreal Engine в обычном режиме, чтобы убедиться, что все по-прежнему работает корректно.

2.2 Инструменты повышения производительности программного обеспечения

Оптимизация структуры уровня: Правильная организация структуры уровня может значительно повлиять на производительность. Разделение уровней на смысловые группы, такие как игровые объекты, земля, здания и т.д., может уменьшить нагрузку на процессор и ускорить процесс загрузки уровня.

Управление ресурсами: Unreal Engine предлагает несколько инструментов для управления ресурсами, таких как текстуры и модели. Одним из них является использование Level of Detail (LOD), что позволяет уменьшить количество полигонов, используемых для объектов в зависимости от расстояния до них. Также, можно использовать сжатие текстур, чтобы уменьшить размер текстур, что может уменьшить затраты на память и ускорить процесс загрузки.

Настройка графических настроек: Уменьшение качества графики может увеличить производительность, особенно на слабых системах. Некоторые параметры, такие как разрешение экрана, графические настройки и теневые настройки, могут быть настроены в зависимости от характеристик системы.

Использование готовых инструментов: Unreal Engine имеет множество готовых инструментов для оптимизации производительности. Один из них – средство профилирования, которое может использоваться для определения бутылочных горлышек в производительности и оптимизации кода. Кроме того, в Unreal Engine есть много плагинов, которые могут помочь ускорить процесс разработки, такие как Mesh Simplification и Lightmap Resolution.

Оптимизация кода: Хороший код может улучшить производительность игры. Один из способов оптимизации кода – это использование алгоритмов с наилучшей сложностью, что может снизить нагрузку на процессор. Также, необходимо избегать повторных вычислений и использовать встроенные функции Unreal Engine, которые уже оптимизированы.

2.3 Средства диагностики оборудования

Для диагностики оборудования при использовании Unreal Engine можно использовать следующие средства:

* Unreal Engine 4 Diagnostic Tools: включает в себя профилирование, отслеживание использования ресурсов, анализ производительности и многое другое. Это инструменты, которые помогают в оптимизации производительности приложения.
* Unreal Engine 4 Hardware Analyzer: инструмент, который позволяет собирать данные об оборудовании и его производительности. Он может использоваться для сбора информации о графическом процессоре, оперативной памяти, процессоре и дисковом пространстве.
* Unreal Engine 4 System Requirements Tool: позволяет проверять, соответствует ли оборудование, на котором запускается приложение, минимальным требованиям Unreal Engine 4.
* Unreal Engine 4 Profiler: инструмент, который позволяет анализировать производительность кода, обнаруживать узкие места и проблемы, связанные с производительностью.
* Unreal Engine 4 Content Examples: набор примеров контента, который может использоваться для проверки и тестирования производительности игры на различных устройствах и конфигурациях.

2.4 Измерение и анализ эксплуатационных характеристик качества программного обеспечения

Для измерения и анализа эксплуатационных характеристик качества Unreal Engine можно использовать различные подходы и инструменты.

Один из основных способов измерения качества Unreal Engine – это тестирование производительности. Это может включать в себя измерение кадровой частоты, задержки ввода, времени отклика, использования памяти, а также других показателей производительности. Для этого можно использовать специальные инструменты, такие как профилировщики, которые позволяют отслеживать производительность и выявлять узкие места в коде.

Другой подход к анализу качества Unreal Engine – это тестирование функциональности. Здесь необходимо проверить, насколько хорошо движок выполняет свои основные функции, такие как визуализация, физика, звук, искусственный интеллект и т.д. Для этого можно использовать различные тесты, такие как тесты на проникновение, функциональное тестирование, тестирование интерфейса пользователя и т.д.

Также для анализа качества Unreal Engine можно использовать методы контроля качества кода. Это включает в себя использование инструментов статического анализа кода, которые позволяют выявлять ошибки в коде до его выполнения, а также использование автоматических тестов, которые позволяют проверить работоспособность кода в автоматическом режиме.

Наконец, для измерения и анализа качества Unreal Engine можно также использовать отзывы пользователей и экспертов, которые могут предоставить ценные отзывы о функциональности и производительности движка. Отзывы пользователей можно собрать на форумах, в социальных сетях и в других онлайн-сообществах, в то время как экспертные мнения можно получить от опытных разработчиков игр и специалистов по разработке программного обеспечения.

2.5 Оптимизация и модификация ПО

Оптимизация и модификация Unreal Engine могут помочь ускорить процесс разработки игры и улучшить ее производительность. Вот несколько советов, которые могут помочь:

Оптимизация настроек движка. Перед началом разработки игры можно настроить многие параметры движка для повышения его производительности и оптимизации его работы.

Использование современных технологий. Unreal Engine поддерживает многие современные технологии, такие как DirectX 12 и Vulkan. Использование этих технологий может улучшить производительность игры.

Использование меньшего количества ресурсов. Unreal Engine может использовать большое количество ресурсов, таких как память и процессорное время. Использование меньшего количества ресурсов может помочь ускорить процесс разработки игры.

Использование более эффективных алгоритмов. Некоторые алгоритмы могут быть более эффективными, чем другие, поэтому использование более эффективных алгоритмов может помочь ускорить процесс разработки игры.

Модификация исходного кода. Unreal Engine имеет открытый исходный код, который можно модифицировать и настроить для удовлетворения конкретных потребностей. Это может помочь улучшить производительность игры и сократить время разработки.

Использование сторонних библиотек и плагинов. Существует множество сторонних библиотек и плагинов, которые можно использовать с Unreal Engine для улучшения его функциональности. Это может помочь сократить время разработки и улучшить производительность игры.

Использование инструментов анализа производительности. Unreal Engine поставляется с инструментами анализа производительности, которые помогают определить узкие места и проблемы в игре. Использование этих инструментов может помочь улучшить производительность игры и сократить время разработки.

3 Этапы сопровождения и обслуживания ПО

3.1 Разработка технического задания на сопровождение

Описание проекта

Проект является игрой на базе Unreal Engine. Игра уже находится в производстве и требуется сопровождение для поддержания ее работоспособности и улучшения ее функциональности.

Требования к сопровождению:

1) Техническая поддержка

Требуется обеспечить техническую поддержку для Unreal Engine, включающую в себя решение проблем, возникающих во время разработки игры, а также настройку и обновление пакетов Unreal Engine.

2) Обновления

Требуется обеспечить обновление игры в соответствии с новыми версиями Unreal Engine, а также обеспечить обновление плагинов, библиотек и других компонентов, необходимых для работы игры.

3) Оптимизация производительности

Требуется производить оптимизацию производительности игры, включая оптимизацию загрузки ресурсов, ускорение процесса загрузки уровней и улучшение работы графических эффектов.

4) Доработка игровых механик

Требуется производить доработку игровых механик, включая создание новых механик и улучшение существующих. Необходимо обеспечивать совместимость доработок с текущей версией игры и Unreal Engine.

5) Тестирование

тестирование игры на различных платформах, включая ПК и мобильные устройства. Тестирование должно включать в себя проверку работоспособности, оптимизации производительности и корректности функционирования игровых механик.

6) Требования к персоналу

Требуется наличие опытных разработчиков, специализирующихся на Unreal Engine, а также опытных тестировщиков для проверки игры. Разработчики и тестировщики должны иметь знания и опыт работы с Unreal Engine и различными платформами.

7) Требования к отчетности

Требуется ведение документации по проекту, включая список изменений и исправлений.

3.2 Обеспечение качества компьютерных систем в процессе эксплуатации

Обеспечение качества компьютерных систем в процессе эксплуатации Unreal Engine – это важный аспект, который необходимо учитывать для обеспечения эффективной работы игрового движка. Вот несколько рекомендаций, которые могут помочь в этом процессе:

* Установка необходимых обновлений: регулярно проверяйте наличие обновлений для вашей операционной системы и Unreal Engine. Обновления часто содержат исправления ошибок и улучшения производительности, поэтому их установка может улучшить стабильность и качество работы системы.
* Новых функций: перед тем, как использовать новые функции Unreal Engine, протестируйте их на локальной машине. Это позволит выявить возможные проблемы и убедиться, что система работает без сбоев.
* Очистка реестра: система реестра может накапливать ошибки и ненужные записи, что может негативно сказаться на производительности и стабильности работы системы. Регулярно используйте утилиты для очистки реестра, чтобы сохранить высокое качество работы системы.
* Очистка жесткого диска: регулярно удаляйте ненужные файлы и программы, чтобы освободить место на жестком диске и улучшить производительность системы.
* Использование антивирусного ПО: установите надежное антивирусное ПО и регулярно проверяйте систему на наличие вредоносных программ. Это поможет предотвратить возможные угрозы для безопасности и сохранить высокое качество работы системы.
* Резервное копирование: регулярно создавайте резервные копии важных данных и файлов, чтобы предотвратить потерю данных в случае сбоя системы или других проблем.
* Мониторинг производительности: используйте программы для мониторинга производительности системы, чтобы выявлять возможные проблемы и оптимизировать работу системы.

В целом, обеспечение качества компьютерных систем в процессе эксплуатации Unreal Engine требует регулярного мониторинга и оптимизации работы системы, чтобы обеспечить максимальную производительность и стабильность при использовании игрового движка.

3.3 Разработка сопровождающей документации

Документацию можно найти на официальном сайте Unreal Engine: <https://docs.unrealengine.com/5.1/en-US/>

4 Методы и средства защиты компьютерных систем

4.1 Тестирование на ошибки ввода/вывода

В процессе работы вы можете сталкнутся с такой проблеммой, как ввод/вывод данных. Чтобы этого избежать, вот примерный план тестирования на ошибки ввода/вывода данных в Unreal Engine:

* Тестирование ввода данных:
* Проверка корректности работы интерфейса ввода данных
* Проверка корректности ввода различных типов данных (целые числа, вещественные числа, строки, булевы значения и т.д.)
* Проверка обработки ошибок при некорректном вводе данных (например, при вводе букв в поле, предназначенное для чисел)
* Проверка корректности ввода данных при использовании различных языков и национальных символов
* Тестирование вывода данных:
* Проверка корректности работы интерфейса вывода данных
* Проверка корректности вывода различных типов данных (целые числа, вещественные числа, строки, булевы значения и т.д.)
* Проверка корректности форматирования выводимых данных
* Проверка корректности вывода данных при использовании различных языков и национальных символов
* Тестирование работы с файлами:
* Проверка возможности открытия и закрытия файлов
* Проверка корректности чтения и записи данных в файлы
* Проверка обработки ошибок при работе с файлами (например, при попытке записи в закрытый файл)
* Проверка корректности работы с различными типами файлов (текстовые, бинарные и т.д.)
* Тестирование работы с сетью:
* Проверка корректности установления и разрыва соединения
* Проверка корректности передачи данных по сети
* Проверка обработки ошибок при работе с сетью (например, при потере соединения)
* Проверка корректности работы с различными протоколами (TCP, UDP и т.д.)
* Тестирование общей работоспособности:
* Проверка корректности обработки ошибок при вводе/выводе данных и работе с файлами и сетью
* Проверка корректности работы программы в целом при использовании различных комбинаций ввода и вывода данных

Конечно, этот список не исчерпывает все возможные сценарии тестирования, и конкретные тесты могут зависеть от конкретных требований к проекту и специфики используемых в нем данных и функциональности.

4.2 Тестирование интерфейса

При работе с Unreal Engine, может возникнуть проблема с интерфейсом. Чтобы таких проблем не происходило, есть список тестов для тестирования интерфейса Unreal Engine:

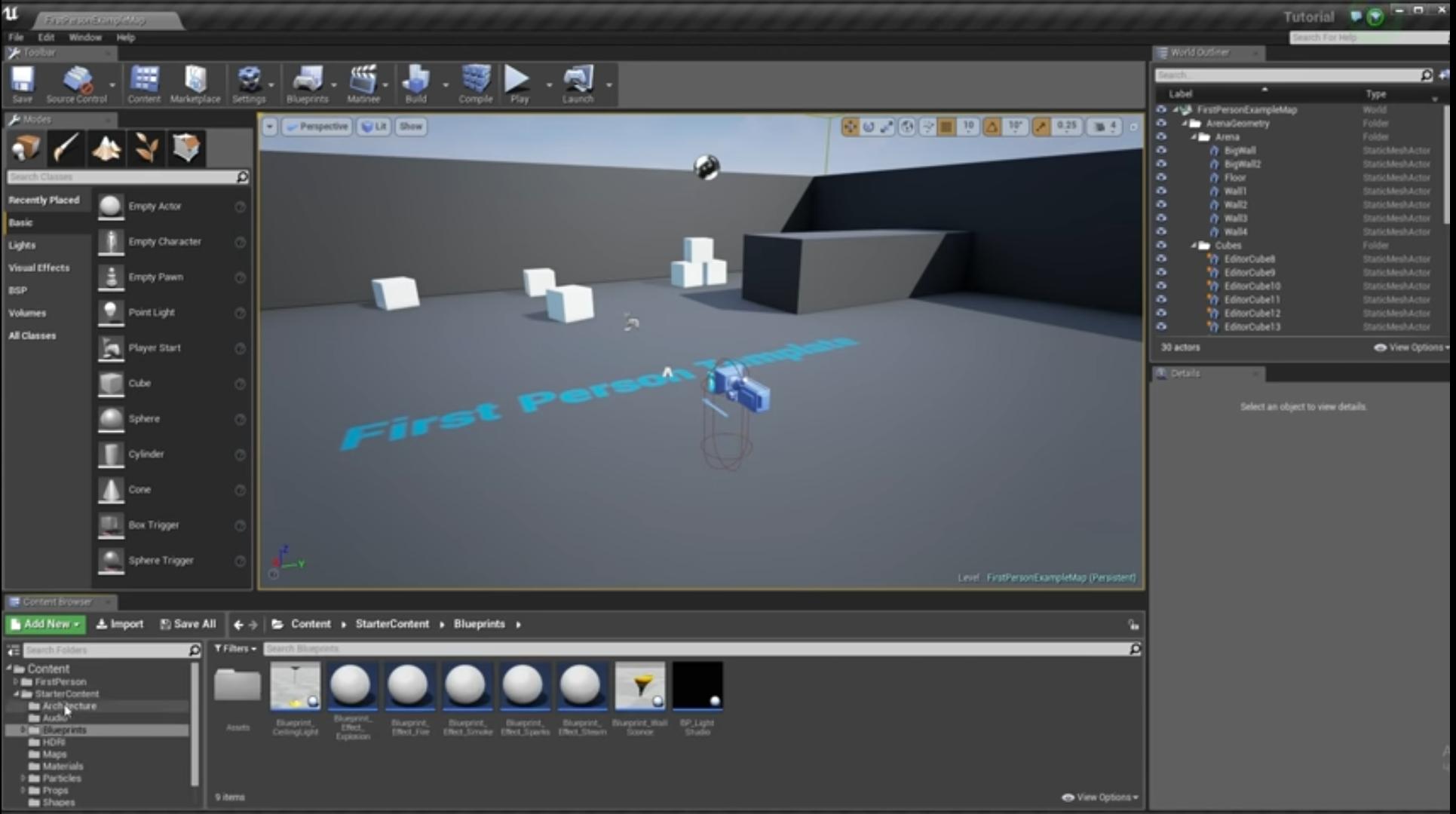


Рис. 6 – Интерфейс Unreal Engine

* Тестирование интерфейса редактора: проверка наличия всех необходимых элементов управления, возможность изменения свойств объектов, функциональность инструментов редактирования и т.д.
* Тестирование возможностей импорта и экспорта данных: проверка возможности импорта различных форматов файлов, правильности отображения данных и экспорта в различные форматы.
* Тестирование производительности: проверка производительности при работе с большими проектами, сценами и множеством объектов.
* Тестирование функциональности материалов и текстур: проверка наличия необходимых инструментов и возможность изменения параметров материалов и текстур, а также правильность отображения результатов.
* Тестирование возможностей работы с сетью: проверка возможности работы в режиме совместной разработки, наличие всех необходимых инструментов для работы с сетевыми подключениями.
* Тестирование работоспособности дополнительных плагинов и расширений: проверка правильности установки и работы с дополнительными плагинами и расширениями, а также их взаимодействия с основной системой Unreal Engine.
* Тестирование документации и помощи: проверка наличия и качества документации, помощи и обучающих материалов, а также их актуальности и полезности.
* Тестирование совместимости: проверка совместимости Unreal Engine с различными операционными системами, аппаратным обеспечением и другими программными продуктами.
* Тестирование безопасности: проверка безопасности работы с Unreal Engine, защиты от вредоносных программ, защиты от несанкционированного доступа и других угроз безопасности.
* Тестирование процесса установки и обновления: проверка установки и обновления Unreal Engine, правильности настройки, актуальности и полноты установочных файлов.

4.3 Тестирование контента

Методы тестирования:

1)Визуальное тестирование моделей 3D:

1.1)Проверка правильности отображения моделей на сцене

1.2)Проверка корректности анимаций и движения моделей

1.3)Проверка столкновений и физики моделей

1.4)Проверка корректности текстур и их отображения на моделях

2)Аудио тестирование звуковых эффектов:

2.1)Проверка корректности воспроизведения звуковых эффектов

2.2)Проверка соответствия звуковых эффектов действиям на сцене

2.3)Проверка громкости и качества звука

3)Тестирование текстур:

3.1)Проверка корректности отображения текстур на моделях

3.2)Проверка соответствия текстур действиям на сцене

3.3)Проверка разрешения и качества текстур

4.4 Анализ рисков. Выявление первичных и вторичных ошибок

Анализ рисков – это процесс выявления, оценки и управления потенциальными рисками, которые могут возникнуть во время разработки или эксплуатации программного обеспечения. Unreal Engine – это игровой движок, который может содержать множество ошибок, как первичных, так и вторичных. В данном контексте первичные ошибки – это те, которые возникают непосредственно в процессе разработки, а вторичные – те, которые возникают в процессе эксплуатации программного обеспечения.

Для проведения анализа рисков в Unreal Engine необходимо выполнить следующие шаги:

* Идентификация потенциальных рисков: определение всех возможных проблем, которые могут возникнуть в процессе разработки и эксплуатации Unreal Engine, включая первичные и вторичные ошибки. Это может быть выполнено путем изучения спецификаций, документации, кода, а также проведения экспертного анализа.
* Оценка рисков: оценка вероятности возникновения каждого риска и его влияния на проект в целом. Это поможет определить, какие риски наиболее важны и нуждаются в первоочередном решении. Оценка рисков может быть выполнена с помощью квалифицированных оценщиков рисков или экспертов в области разработки Unreal Engine.
* Разработка стратегии управления рисками: разработка плана действий по управлению каждым риском в случае его возникновения. В этом плане могут быть указаны шаги, которые нужно предпринять для снижения влияния риска, а также уменьшения вероятности его возникновения.
* Реализация плана действий: выполнение плана по управлению рисками и отслеживание его реализации. Это может включать устранение первичных ошибок, мониторинг работы Unreal Engine во время эксплуатации и анализ вторичных ошибок, которые могут возникнуть.
* Мониторинг рисков: постоянное отслеживание состояния рисков и управление ими в зависимости от изменяющихся условий в проекте.

В итоге, проведение анализа рисков позволяет выявить и устранить потенциальные проблемы, которые могут возникнуть в процессе разработки и эксплуатации Unreal Engine

4.5 Обнаружение вируса и устранение последствий его влияния

При установке неизвестного ПО с интернета, есть вероятность подхватить врус. Который в своё очередь, может сильной повлиять на процесс создания игры. Чтобы такого не происходило, вот несколько шагов, которые могут помочь в этом процессе:

1)Обнаружение вируса:

Сначала необходимо убедиться, что ваш компьютер защищен антивирусным программным обеспечением. Проверьте, была ли установлена и обновляется ли антивирусная программа на вашем компьютере. Запустите полное сканирование системы антивирусной программой для обнаружения любых возможных вирусов. Если антивирусная программа обнаружит угрозы, следуйте ее инструкциям для удаления вирусов.

Если антивирусная программа не обнаружила вирусов, попробуйте использовать антишпионское программное обеспечение, чтобы обнаружить любые вредоносные программы или шпионские программы.

2)Устранение последствий влияния вируса:

* Перезагрузите компьютер и запустите Unreal Engine.
* Проверьте, работает ли Unreal Engine должным образом, и если есть какие-либо проблемы, попробуйте выполнить следующие действия:
* Отключите все плагины и расширения, установленные в Unreal Engine.
* Удалите и повторно установите Unreal Engine.
* Проверьте, что ваша операционная система и драйверы обновлены до последней версии.
* Проверьте файлы настроек Unreal Engine на наличие изменений.
* Если все еще есть проблемы, связанные с Unreal Engine, обратитесь к технической поддержке Epic Games или к сообществу Unreal Engine для получения дополнительной помощи.

3)Профилактические меры:

* Убедитесь, что ваш компьютер защищен антивирусным программным обеспечением и всегда обновляется до последней версии.
* Не скачивайте и не устанавливайте программное обеспечение с ненадежных или неизвестных источников.
* Регулярно проверяйте компьютер на наличие вирусов и других вредоносных программ.
* Не открывайте электронные письма и вложения от незнакомых отправителей.
* Не принимайте подозрительные запросы на установку дополнительных программ или расширений.

4.6 Тестирование защиты программного обеспечения

Также не стоит забывать о тестировании защиты Unreal Engine. Вот несколько пунктов, которые можно использовать для тестирования защиты Unreal Engine:

* Проверка контроля целостности файлов – убедитесь, что все файлы, используемые в игре, подписаны и защищены от взлома.
* Анализ потенциальных уязвимостей – проведите анализ на предмет возможных уязвимостей, которые могут использоваться злоумышленниками для взлома игры.
* Тестирование противодействия взлому – используйте инструменты для тестирования противодействия взлому, такие как внедрение вредоносного кода и анализ попыток взлома.
* Анализ использования защитных механизмов – убедитесь, что все защитные механизмы Unreal Engine используются в игре, и что они правильно настроены.
* Тестирование защиты от обратного инжиниринга – проведите тестирование на предмет возможности извлечения кода игры или другой интеллектуальной собственности из игры.
* Проверка обновлений безопасности – убедитесь, что все обновления безопасности для Unreal Engine установлены и настроены правильно

4.7 Работа с реестром. Работа с программой восстановления файлов и очистки дисков

Каждый, кто знаком с работой Windows Explorer, будет чувствовать себя уютно при работе со Starus File Recovery. Программа отобразит содержимое диска, показывая удаленные файлы и папки на тех местах, где они располагались до удаления. Встроенная функция предварительного просмотра поддерживает сотни файловых форматов.

Starus File Recovery работает во всех версиях Windows, включая Windows 7 и Windows 2008 Server, и поддерживает все файловые системы, включая FAT16, FAT32, NTFS4 и NTFS5. Программа также восстанавливает сжатые, шифрованные файлы NTFS и альтернативные потоки, прикрепленные к файлам.

Скачать его можно здесь: <https://www.starusrecovery.ru/file_recover/software.html>

5 Индивидуальное задание [9]

5.1 Практические работы

Практическая работа №1 «Внедрение программного обеспечения»

Внедрение программного обеспечения — процесс настройки программного обеспечения под определенные условия использования, а также обучение пользователей работе с программным продуктом.

Внедрение программного продукта состоялось в том случае, если программный продукт выполняет поставленную задачу, а сотрудники компании полностью перешли на работу с новым продуктом.

Результатом проекта внедрения должен быть прозрачный, четко регламентированный, документированный и автоматизированный процесс разработки и сопровождения.

Основные этапы внедрения программного продукта:

1. Обследование

2. Разработка технического задания

3. Настройка системы

4. Тестирование системы

5. Опытная эксплуатация

6. Промышленная эксплуатация

Эффекты от внедрения можно разделить на две большие категории: эффект, получаемый на стратегическом уровне, т.е. на уровне бизнеса, и тактический эффект, который получат все участники процесса.

1: Стратегический эффект

* обеспечение прозрачности и измеряемости достижения стратегических целей;
* снижение вероятности проявления рисков в сфере информационных технологий;
* повышение рентабельности IT-услуг за счет снижения сроков проведения проекта, снижения издержек поддержки и т.д.;
* повышение инвестиционной привлекательности IT-проектов;
* повышение доверия бизнеса к IT за счет качественно обработанных запросов IT-отделами от бизнеса.

2: Тактический эффект

* уменьшение сроков и снижение стоимости обработки новых запросов;
* улучшение качества IT-услуг – это вовремя выполненные работы с заранее определенным качеством;
* увеличение эффективности используемых ресурсов – хорошо организованный и формализованный процесс позволяет четко определить роли и ответственности каждого участника этого процесса;
* более четкое и реалистичное планирование;
* значительное уменьшение времени на принятие решения;
* снижение влияния человеческого фактора.

Сопровожде́ние (поддержка) программного обеспечения — процесс улучшения, оптимизации и устранения дефектов программного обеспечения (ПО) после передачи в эксплуатацию.

Сопровождение ПО — это одна из фаз жизненного цикла программного обеспечения, следующая за фазой передачи ПО в эксплуатацию.

Определение процесса сопровождения:

* SWEBOK: Сопровождение ПО – вся совокупность деятельности, необходимой для обеспечения эффективной поддержки программных систем.
* IEEE 1219 (Standard for Software Maintenance): Сопровождение ПО – определяется стандартом IEEE Standard for Software Maintenance (IEEE 1219) как модификация программного продукта после передачи в эксплуатацию для устранения сбоев, улучшения показателей производительности и/или других характеристик (атрибутов) продукта, или адаптации продукта для использования в модифицированном окружении.
* ГОСТ Р ИСО/МЭК 12207: Сопровождение – как процесс модификации программного продукта в части его кода и документации для решения возникающих проблем при эксплуатации или реализации потребностей в улучшениях тех или иных характеристик продукта.

Сопровождение поддерживает функционирование программного продукта на протяжении всего операционного жизненного цикла, то есть периода его эксплуатации.

В процессе сопровождения фиксируются и отслеживаются запросы на модификацию (также называемые “запросами на изменения” – change requests, в частности, в контексте конфигурационного управления), оценивается влияние предлагаемых изменений, производится модификация кода и других активов (артефактов) продукта, проводится необходимое тестирование и, наконец, выпускается обновленная версия продукта. Кроме того, проводится обучение пользователей и обеспечивается их ежедневная поддержка при работе с текущей версией продукта.

Практическая работа №2 «Техническая поддержка ПО»

Образец договора технической поддержки программного обеспечения.

1. Термины договора на сопровождение программ

1.1. Продукты – предоставленное по лицензионному договору № \_\_ от «\_\_»\_\_\_\_\_\_\_\_\_\_\_ 202\_ г. Программное обеспечение.

1.2. Техническая поддержка – оказываемые услуги по настройке, обслуживанию, адаптации и модификации Продуктов или устранению имеющихся в них ошибок, а также предоставлению обновлений и дополнительных программных модулей, иные действия, предусмотренные в разделе 2 настоящего Договора.

1.3. Ошибка – дефект в коде Продукта, в результате которого данный Продукт не способен работать в соответствии функциональными возможностями, указанными в предоставленной на него технической документации, за исключением случаев:

(1) нарушения Заказчиком правил эксплуатации Продуктов в соответствии с требованиями предоставленной на них технической документации;

(2) использования Продуктов на оборудовании или совместно с программным обеспечением, которые не были рекомендованы Исполнителем.

2. Предмет договора техподдержки ПО

2.1. Исполнитель обязуется оказывать по заявкам Заказчика услуги по Технической поддержке, а Заказчик принимать и оплачивать оказываемые Исполнителем услуги.

2.2. Стороны согласовали возможность оказания услуг по Технической поддержке в следующем объеме и составе:

* установка Продуктов на оборудование Заказчика;
* настройка Продуктов на оборудовании Заказчика, включая их адаптацию;
* модификация Продуктов по отдельному заказу;
* предоставление выпускаемых обновлений Продуктов;
* устранение ошибок в Продуктах;
* консультирование по порядку использования Продуктов.

2.3. Услуги по Договору оказываются в соответствии с Соглашением об уровне услуг (SLA), являющимся неотъемлемой частью настоящего Договора.

Практическая работа №3 «Защита ПО»

Методы доказательства правильности программ могут быть применены для анализа безопасности ПО при существенных ограничениях на размеры и сложность создаваемых программ. Поэтому в частных случаях они могут оказаться более эффективными, чем другие известные методы анализа программ, которые исследуются в следующих разделах данной работы.

Методы, используемые для анализа и оценки безопасности ПО, разделяют на две категории: контрольно-испытательные и логико-аналитические. В основу данного разделения положены принципиальные различия в точке зрения на исследуемый объект (программу).

Контрольно-испытательные методы анализа рассматривают РПС через призму фиксации факта нарушения безопасного состояния системы, а логико-аналитические – через призму доказательства наличия отношения эквивалентности между моделью исследуемой программы и моделью РПС.

Контрольно-испытательные делятся на те, в которых контролируется процесс выполнения программы и те, в которых отслеживаются изменения в операционной среде, к которым приводит запуск программы.

При проведении анализа безопасности с помощью логико-аналитических методов строится модель программы и формально доказывается эквивалентность модели исследуемой программы и модели РПС. В простейшем случае в качестве модели

В целом полный процесс анализа ПО включает в себя три вида анализа:

* + лексический верификационный анализ;
  + синтаксический верификационный анализ;
  + семантический анализ программ

Каждый из видов анализа представляет собой законченное исследование программ согласно своей специализации.

Результаты исследования могут иметь как самостоятельное значение, так и коррелироваться с результатами полного процесса анализа.

Лексический верификационный анализ предполагает поиск распознавания и классификацию различных лексем объекта исследования (программа), представленного в исполняемых кодах. При этом лексемами являются сигнатуры. В данном случае осуществляется поиск сигнатур следующих классов:

* сигнатуры вирусов;
* сигнатуры элементов РПС;
* сигнатуры (лексемы) «подозрительных функций»;
* сигнатуры штатных процедур использования системных ресурсов и внешних устройств.

Поиск лексем (сигнатур) реализуется с помощью специальных программ-сканеров.

Синтаксический верификационный анализ предполагает поиск, распознавание и классификацию синтаксических структур РПС, а также по-строение структурно-алгоритмической модели самой программы.

Решение задач поиска и распознавания синтаксических структур РПС имеет самостоятельное значение для верификационного анализа программ, поскольку позволяет осуществлять поиск элементов РПС, не имеющих сигнатуры. Структурно-алгоритмическая модель программы необходима для реализации следующего вида анализа – семантического.

Семантический анализ предполагает исследование программы изучения смысла составляющих ее функций (процедур) в аспекте операционной среды компьютерной системы. В отличие от предыдущих видов анализа, основанных на статическом исследовании, семантический анализ нацелен на изучение динамики программы – ее взаимодействия с окружающей средой. Процесс исследования осуществляется в виртуальной операционной среде с полным контролем действий программы и отслеживанием алгоритма ее работы по структурно-алгоритмической модели.

Семантический анализ является наиболее эффективным видом анализа, но и самым трудоемким. По этой причине методика сочетает в себе три перечисленных выше анализа. Выработанные критерии позволяют разумно сочетать различные виды анализа, существенно сокращая время исследования, не снижая его качества.

Практическая работа №4.

|  |  |
| --- | --- |
| Тип комплектующих | Модель комплектующих |
| Процессор | Intel Core i7-11700k |
| Оперативная память | 8 GB DDR3 |
| Видеокарта | Radion RX 580 8 gb |
| Звуковая плата | Интегрированный контроллер |
| Сетевая карта | Интегрированный контроллер |
| Жёсткий диск | 500gb, 250gb |
| Дисковод для гибких дисков | 25 x1, 3,5 дюйма x2. |
| Оптический привод | - |
| Клавиатура | PANTEON M2 |
| Мышь | Defender Sleipnir |
| Монитор | AOC M2470SWD2 |
| Корпус | Ginzzu CL180 |
| Блок питания | 600W |
| Источник бесперебойного питания | - |

5.2 Порядок установки и сопровождения клиентского программного обеспечения

Ниже приведен порядок установки и сопровождения клиентского программного обеспечения Unreal Engine:

1) Unreal Engine:

* Скачайте установщик Unreal Engine с официального сайта.
* Запустите установщик и следуйте инструкциям по установке, включая выбор нужных компонентов и места установки.
* После завершения установки запустите Unreal Engine и зарегистрируйтесь в Epic Games, чтобы получить доступ к последним обновлениям и инструментам.

2)Настройка Unreal Engine:

* Настройте параметры проекта, включая настройку режимов отладки, установку ресурсов и другие параметры.
* Установите параметры безопасности, такие как автоматические обновления и настройки доступа.

3)Сопровождение Unreal Engine:

* Регулярно обновляйте Unreal Engine, чтобы получить последние исправления ошибок и новые функции.
* Отслеживайте обновления плагинов и дополнительных ресурсов, которые могут быть полезными для вашего проекта.
* Используйте документацию Unreal Engine, форумы и другие ресурсы, чтобы получить помощь и решить проблемы.

5.3 Практическая работа №9 «Связь сопровождения с эволюцией программного обеспечения»

Unreal Engine – это игровой движок, разработанный компанией Epic Games. Компания Epic Games была основана Тимом Суини в 1991 году под названием Potomac Computer Systems. В начале своей деятельности компания занималась разработкой игр для персональных компьютеров, в том числе ZZT и Jill of the Jungle. В 1998 году компания переименовалась в Epic MegaGames и выпустила свою первую игру на новом движке - Unreal. В 1999 году компания снова изменила свое название, став Epic Games. Первая версия Unreal Engine была выпущена в 1998 году и была предназначена для создания игр от первого лица, таких как Unreal и Unreal Tournament. С тех пор Unreal Engine был значительно усовершенствован и развивается до сих пор.

Сопровождение программного обеспечения, включая Unreal Engine, является важным аспектом разработки программных продуктов. Оно включает в себя постоянное обновление и поддержку продукта после его выпуска. Это включает в себя исправление ошибок, обновление функций и улучшение производительности.

Сопровождение связано с эволюцией программного обеспечения, так как с каждым обновлением Unreal Engine улучшается и приспосабливается к новым требованиям и технологиям. Например, с выпуском новых версий Unreal Engine добавляются новые возможности, такие как улучшенная поддержка виртуальной реальности, улучшенная графика и физика, а также инструменты для создания многопользовательских игр.

|  |  |
| --- | --- |
| Версия движка | Краткое описание |
| Unreal Engine 1 | Выпущенный в 1998 году, первый версия движка, который использовался для создания игры Unreal. Движок предоставлял возможности для рендеринга графики, создания и управления игровыми объектами, а также работы с звуком. |
| Unreal Engine 2 | Выпущенный в 2002 году, вторая версия движка, который использовался для создания игр на платформах PlayStation 2, Xbox и PC. Вторая версия Unreal Engine добавила поддержку динамического освещения, улучшенные возможности для работы с звуком и новые инструменты для создания и анимации персонажей. |
| Unreal Engine 3 | Выпущенный в 2006 году, третья версия движка, который использовался для создания игр на PlayStation 3, Xbox 360 и PC. Он предоставлял множество новых возможностей для разработки игр, включая более продвинутый движок физики, поддержку DirectX 10 и улучшенную систему освещения. |
| Unreal Engine 4 | Выпущенный в 2014 году, четвертая и наиболее продвинутая версия Unreal Engine, доступная для широкой аудитории. Unreal Engine 4 предоставляет разработчикам мощные инструменты для создания высококачественных игр на платформах PC, PlayStation, Xbox, Nintendo Switch и многих других. Он также имеет встроенную поддержку виртуальной реальности, улучшенные возможности для создания и управления игровыми мирами, а также инструменты для работы с анимациями, звуком и физикой. |
| Unreal Engine 5 | Был представлен в 2020 году и еще не выпущен официально. Он обещает предоставить еще больше возможностей для создания высококачественных игр с мощным движком рендеринга Nanite и инструментами для работы с геометрическими формами. |

Начиная с версии Unreal Engine 4, компания Epic Games решила сделать движок бесплатным для использования, а доходы получать за счет продажи лицензий на инструменты и услуги разработки игр, созданных с помощью Unreal Engine. Это привело к значительному увеличению числа пользователей Unreal Engine.

Важным аспектом сопровождения Unreal Engine является обратная совместимость между различными версиями движка. Это позволяет разработчикам игр сохранять совместимость между различными версиями движка и обеспечивать поддержку своих игр в течение длительного времени. Например, версия Unreal Engine 4.26 была выпущена в декабре 2020 года и до сих пор обновляется для исправления ошибок и улучшения производительности.

Кроме того, Epic Games постоянно добавляет новые функции и инструменты в Unreal Engine. Например, в версии 4.27, выпущенной в сентябре 2021 года, были добавлены новые инструменты для работы с анимациями, улучшена поддержка платформы PlayStation 5, а также добавлены новые функции для работы с виртуальной реальностью.

Еще одним важным аспектом сопровождения Unreal Engine является поддержка сообщества разработчиков, которое постоянно общается с Epic Games и предлагает свои идеи и улучшения для движка. Кроме того, Epic Games регулярно проводит обучающие курсы и вебинары, чтобы помочь новым пользователям быстрее освоить Unreal Engine и научиться использовать его наиболее эффективно.

Сопровождение Unreal Engine является неотъемлемой частью его эволюции и развития, что позволяет разработчикам создавать все более продвинутые и высококачественные игры. Благодаря постоянным обновлениям и улучшениям Unreal Engine остается одним из наиболее популярных игровых движков в мире.

Заключение

Внедрение Unreal Engine – это значимое решение для разработки компьютерных игр и других визуальных проектов в различных отраслях. Он предоставляет мощный инструментарий для создания реалистичных и захватывающих игровых миров, а также для создания виртуальных симуляций, тренировок и визуализаций для обучения, науки и инженерии.

Одним из главных преимуществ Unreal Engine является его возможность создания кроссплатформенных приложений для различных устройств и операционных систем. Он также поддерживает большое количество различных языков программирования, что делает его очень гибким и доступным для широкого круга разработчиков.

Помимо этого, Unreal Engine обладает широкой поддержкой сообщества, которое активно создает и обменивается инструментами, ресурсами и знаниями. Это создает благоприятную среду для развития проектов и обучения новым навыкам.

В целом, внедрение Unreal Engine может повысить качество и производительность разрабатываемых проектов, сократить время разработки и расширить возможности создания и визуализации уникальных и интересных игровых миров и других визуальных проектов.

Список используемых источников

Официальный Unreal Engine: <https://www.unrealengine.com/>

Unreal Engine Wiki: <https://en.wikipedia.org/wiki/Unreal_Engine>

Unreal Engine ПК: <https://delta-game.ru/news/kompyuter-dlya-unreal-engine/>

Unreal Engine ПК: <https://blog.skillfactory.ru/glossary/unreal-engine/>

Unreal Engine история: <https://meownauts.com/unreal-engine-history/>

Unreal Engine для новичков: https://dtf.ru/unrealengine/1300597-sovety-dlya-nachinayushchih-i-ne-ochen-razrabotchikov-na-unreal-engine

Документация Unreal Engine: https://docs.unrealengine.com/5.1/en-US/

Уроки по Unreal Engine:

https://m.youtube.com/channel/UCBobmJyzsJ6Ll7UbfhI4iwQ

Фондом: https://gamegod.fandom.com/ru/wiki/Unreal\_Engine

Книга «Разработка игр на Unreal Engine 4 за 24 часа»

GitHub https://github.com/4amer/-04.git