

Всероссийское СМИ

«Академия педагогических идей «НОВАЦИЯ»

Свидетельство о регистрации Эл №ФС 77-62011 от 05.06.2015 г.

(выдано Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций)

Сайт: akademnova.ru

e-mail: akademnova@mail.ru

Акшенцев М.А., Григорьев М.Г., Кильмакаев В.А. Многоеритериальный выбор игровых движков для создания 8-ми, 16-ти битной 2D RPG игры // Академия педагогических идей «Новация». Серия: Студенческий научный вестник. – 2026. – №4 (май) – АРТ 10-эл. – 0,3 п.л. - URL: <http://akademnova.ru/page/875550>

РУБРИКА: ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

УДК 004.9

Акшенцев Михаил Андреевич,

студент 1 курса, институт информатики, математики и робототехники

Григорьев Максим Григорьевич,

студент 1 курса, институт информатики, математики и робототехники

Кильмакаев Вадим Аликович,

студент 1 курса, институт информатики, математики и робототехники

Научный руководитель: Воробьев А.В., профессор, д.т.н

ФГБОУ ВО «Уфимский университет науки и технологий»

г. Уфа, Российская Федерация

e-mail: kilmakaevvadim@gmail.com

МНОГОЕРИТЕРИАЛЬНЫЙ ВЫБОР ИГРОВЫХ ДВИЖКОВ ДЛЯ СОЗДАНИЯ 8-МИ , 16-ТИ БИТНОЙ 2D RPG ИГРЫ

Аннотация: В статье рассматривается процесс разработки компьютерной игры с использованием игрового движка Unity. Проведён сравнительный анализ игровых движков Unity, Unreal Engine и Godot. Обоснован выбор Unity как основного инструмента разработки. Описаны этапы создания игры, архитектура программного продукта, а также трудности, возникающие в процессе разработки.

Всероссийское СМИ

«Академия педагогических идей «НОВАЦИЯ»

Свидетельство о регистрации Эл №ФС 77-62011 от 05.06.2015 г.

(выдано Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций)

Сайт: akademnova.ru

e-mail: akademnova@mail.ru

Ключевые слова: Unity, разработка игр, игровой движок, C#, UML, архитектура программного обеспечения.

Akshentsev Mihail,

1nd year students, Institute of Informatics, Mathematics and Robotics

Grigoriev Maxim,

1nd year students, Institute of Informatics, Mathematics and Robotics

Kilmakayev Vadim,

1nd year students, Institute of Informatics, Mathematics and Robotics

Supervisor: Vorobiev A.V., Professor, Doctor of Technical Sciences

FGBOU VPO “Ufa University of Science and Technology”

Ufa, Russian Federation

e-mail: kilmakaevvadim@gmail.com

A MULTI-MATERIAL SELECTION OF GAME ENGINES FOR CREATING 8-BIT, 16-BIT 2D RPG GAMES

Abstract: The article describes the process of developing a computer game using the Unity game engine. A comparative analysis of Unity, Unreal Engine and Godot game engines is carried out. The choice of Unity as the main development tool is justified. The stages of game creation, software architecture, as well as difficulties arising during development are described.

Keywords: Unity, game development, game engine, C#, UML, software architecture.

Введение

Цель – анализ процесса разработки игры и обоснование выбора игрового движка Unity.

Современная игровая индустрия является одной из наиболее динамично развивающихся областей информационных технологий. Компьютерные игры используются не только для развлечения, но и в образовательных целях, моделировании процессов и развитии логического мышления.

В рамках данной работы рассматривается разработка компьютерной игры, представляющей собой сочетание жанров RPG и визуальной новеллы. Особенностью проекта является разделение игрового процесса на два мира: реальный и мир снов.

Многокритериальное сравнения альтернатив

В качестве альтернатив рассматриваются игровые движки для разработки 2D RPG-игры. Когда имеется множество игровых движков для решения одной и той же задачи, возникает необходимость их специального анализа с целью обоснованного выбора (например, [5]).

Критерии оценки игровых движков

Для проведения анализа были определены следующие критерии, отражающие как технические, так и пользовательские аспекты использования игровых движков:

К₁. Язык программирования.

К₂. Уровень графики.

К₃. Сложность освоения.

К₄. Кроссплатформенность.

К₅. Сообщество.

К₆. Asset Store.

Сравнительный анализ игровых движков

На основе собственного опыта разработчиками были оценены игровые движки по каждому критерию (табл. 1).

Таблица 1

Критериальные оценки движков для разработки 2D игры

Критерии	Unity	Unreal Engine	Godot
K ₁	C#	C++ / Blueprints	GDScript
K ₂	Средний / высокий	Очень высокий	Средний
K ₃	Низкая	Высокая	Средняя
K ₄	Высокая	Высокая	Высокая
K ₅	Очень большое	Большое	Среднее
K ₆	Развитый	Ограниченный	Ограниченный

Результаты анализа и обоснование выбора

Unreal Engine обеспечивает наилучшее качество графики, однако требует значительных ресурсов и глубоких знаний программирования.

Godot является легковесным и бесплатным решением, однако уступает по функциональности и количеству готовых инструментов.

Unity сочетает простоту освоения, широкие возможности и развитую экосистему, что делает его оптимальным выбором для разработки инди-игр [1; 2; 4].

Выбор Unity для данного проекта обусловлен следующими ключевыми преимуществами:

- Поддержка множества платформ.

Unity позволяет создавать игры, которые могут быть запущены на самых разных платформах, включая настольные операционные системы (Windows, macOS, Linux), мобильные устройства (iOS, Android), игровые консоли (PlayStation, Xbox, Nintendo Switch) и веб-браузеры. Это значительно расширяет потенциальную аудиторию игры и упрощает процесс портирования на различные устройства.

– Использование языка C#.

C# – это современный, объектно-ориентированный язык программирования, который широко используется в разработке игр [1; 2; 6]. Он обладает относительно простым синтаксисом, мощными возможностями и отличной поддержкой со стороны Unity. Разработчики, знакомые с C#, могут быстро приступить к работе, а для новичков язык предлагает хорошую кривую обучения.

– Unity Asset Store.

Unity Asset Store – это обширный магазин, где разработчики могут приобрести или скачать бесплатно готовые ресурсы [9]: 3D-модели, текстуры, звуки, скрипты, шаблоны и инструменты. Это позволяет значительно ускорить процесс разработки, особенно для инди-студий или проектов с ограниченным бюджетом, так как не требует создания всего контента с нуля.

– Развитая документация и сообщество.

Unity имеет одну из самых полных и хорошо организованных документаций среди игровых движков [9]. Кроме того, существует огромное и активное глобальное сообщество разработчиков. Это означает, что при возникновении трудностей или вопросов, всегда можно найти ответы в официальных источниках, на форумах, в обучающих материалах и на специализированных сайтах.

- Визуальный редактор сцен.

Unity предоставляет интуитивно понятный визуальный редактор, который позволяет дизайнерам и разработчикам создавать и настраивать игровые сцены, размещать объекты, настраивать освещение и камеры, не прибегая к написанию кода. Это значительно упрощает и ускоряет процесс прототипирования и итераций в дизайне уровней.

Недостатки Unity.

Несмотря на многочисленные преимущества, Unity имеет и свои ограничения:

- Сравнительно высокая нагрузка на систему.

Unity может потреблять значительные системные ресурсы как во время разработки (работа редактора), так и во время выполнения игры. Это может потребовать более мощного "железа" для комфортной работы и оптимизации игры под менее производительные устройства.

- Ограничения в создании фотореалистичной графики.

Хотя Unity постоянно развивается в области графики, достижение уровня фотореализма, сравнимого с некоторыми специализированными движками (например, Unreal Engine), может потребовать значительных усилий, глубоких знаний и использования сторонних инструментов или продвинутых техник рендеринга.

Процесс разработки игры

Разработка игры – это многоэтапный процесс, включающий планирование, проектирование, реализацию и тестирование [3; 7; 8]. В данном проекте были пройдены следующие этапы:

Проектирование

На начальном этапе был проведен анализ требований к игре и определена её основная концепция. Для формализации архитектуры системы

и определения взаимосвязей между её компонентами были разработаны UML-диаграммы. Эти диаграммы служили "дорожной картой" для дальнейшей разработки, обеспечивая четкое понимание структуры проекта.

Архитектура

Система была спроектирована с использованием модульного принципа, что обеспечивает гибкость, расширяемость и упрощает поддержку кода. Каждый модуль отвечает за определенную функциональность игры. Основные модули включают:

- GameManager.

Отвечает за общее управление игровым процессом, состоянием игры, загрузкой и сохранением данных, а также координацию работы других менеджеров.

- DialogueManager.

Управляет всей системой диалогов, включая отображение текста, выбор реплик игроком, ветвление сюжета и взаимодействие с NPC.

- CombatSystem.

Реализует механику боевой системы, включая управление персонажами в бою, расчет урона, применение навыков и эффектов, а также логику победы/поражения.

- RelationshipSystem.

Обработывает систему отношений между персонажами. Отвечает за изменение уровня дружелюбия/враждебности, влияние отношений на диалоги и сюжетные события.

- UIManager.

Отвечает за управление всеми пользовательскими интерфейсами игры: меню, HUD, окна диалогов, инвентарь и т.д.

Реализация

- Программирование.

Основное программирование выполнено на языке C#, с использованием возможностей движка Unity [1; 2; 6].

- Разделение игрового процесса.

Игровой процесс был условно разделен на два основных "мира":

- Реальный мир: включает в себя исследование игрового мира, взаимодействие с NPC, выполнение заданий и ведение диалогов. Здесь основное внимание уделяется повествованию и развитию персонажей.

- Мир снов: представляет собой боевую систему игры. Переход в этот мир может быть инициирован определенными событиями или сюжетными поворотами, и здесь игрок сталкивается с противниками в пошаговых или динамических сражениях.

Тестирование

На протяжении всего процесса разработки проводилось тестирование для обеспечения качества и стабильности игры:

- Модульное тестирование.

Проверка работоспособности каждого отдельного модуля (компонента) системы. Позволяет выявить и исправить ошибки на самом раннем этапе.

- Функциональное тестирование.

Проверка того, как различные части игры взаимодействуют друг с другом и соответствуют ли они заявленным требованиям. Это тестирование охватывает игровые механики, пользовательский интерфейс и сюжетные линии.

Трудности разработки

В процессе разработки были выявлены и успешно преодолены следующие проблемы:

- Сложность взаимодействия между модулями.

При построении сложной системы с множеством взаимосвязанных модулей возникает задача обеспечения их эффективного и надежного взаимодействия. Было решено с помощью четко определенных интерфейсов и паттернов проектирования, минимизирующих прямые зависимости.

- Реализация системы анимаций.

Создание плавной и правдоподобной анимации персонажей, особенно в контексте различных игровых состояний (ходьба, бег, бой, диалоги), требует значительных усилий и понимания инструментов анимации Unity.

- Разработка боевой системы.

Проектирование сбалансированной, интересной и технически реализованной боевой системы является одной из самых сложных задач [3; 7; 8]. Это включает в себя механику атаки, защиты, применение умений, искусственный интеллект противников и балансировку сложности.

- Создание нелинейной диалоговой системы.

Разработка системы, позволяющей игроку влиять на сюжет через выбор реплик, создавать множество веток диалогов и последствий этих выборов, является трудоемкой задачей, требующей тщательного планирования и организации сценария.

- Обеспечение стабильности работы приложения.

Поддержание стабильной работы игры на различных устройствах, предотвращение сбоев, утечек памяти и других ошибок – это постоянная задача, требующая внимательности к деталям и регулярного тестирования.

Заключение

В результате была разработана компьютерная игра с использованием Unity. Проведенный анализ показал, что данный движок является оптимальным инструментом для разработки инди-проектов.

Всероссийское СМИ

«Академия педагогических идей «НОВАЦИЯ»

Свидетельство о регистрации Эл №ФС 77-62011 от 05.06.2015 г.

(выдано Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций)

Сайт: akademnova.ru

e-mail: akademnova@mail.ru

В дальнейшем проект может быть расширен за счет добавления новых механик и улучшения визуальной составляющей.

Список используемой литературы:

1. Бонд Д. Г. Unity и C#. Геймдев от идеи до реализации : пер. с англ. / предисл. Р. Лемарчанда. 2-е изд. СПб. [и др.] : Питер, 2025. 925 с.
2. Ferrone H. Learning C# by Developing Games with Unity 6. 8th ed. Birmingham : Packt Publishing, 2025.
3. Rogers S. Level Up! The Guide to Great Video Game Design. 3rd ed. Hoboken : John Wiley & Sons, Inc., 2025.
4. Järvenpää S. Capabilities of the Open-Source Godot Engine in Game Development Compared to Unity and Unreal Engine : Bachelor's thesis. Tampere University, 2025. URL: <https://trepo.tuni.fi/handle/10024/227088> (дата обращения: 16.05.2026).
5. Pelimoottoreiden vertailu : Unity, Unreal Engine ja vaihtoehtoiset ratkaisut // OuluREPO. 2025. URL: <https://oulurepo oulu.fi/handle/10024/57646> (дата обращения: 16.05.2026).
6. Farmer M. The C# Playbook: Essential Skills and Projects for .NET and Unity Developers. Independently Published, 2025.
7. Kocurek C. A., Payne M. T. Ultima and Worldbuilding in the Computer Role-Playing Game. Amherst : Amherst College Press, 2024.
8. Wainwright M., Ho P.-I. The Legacy of Tabletop Role-Playing Games: Shaping Contemporary Game Design // Proceedings of DiGRA Australia 2024. Digital Games Research Association Australia, 2024. URL: <https://digraa.org/wp-content/uploads/2024/02/DiGRAA24-15-Wainwright-Ho.pdf> (дата обращения: 16.05.2026).
9. Junior Programmer Pathway – Learn Coding & Game Development // Unity Learn. URL: <https://learn.unity.com/pathway/junior-programmer> (дата обращения: 16.05.2026). © Акшенцев М.А., Григорьев М.Е., Кильмакаев В.А. 2026

Дата поступления в редакцию: 17.05.2026 г.

Опубликовано: 18.05.2026 г.

© Академия педагогических идей «Новация».

Серия «Студенческий научный вестник», электронный журнал, 2026

© Акшенцев М.А., Григорьев М.Г., Кильмакаев В.А., 2026