

Чжан Цзэхао, магистрант,
Амурский государственный университет

ИССЛЕДОВАНИЕ КОМПЛЕКСНОГО ПРОИЗВОДСТВЕННОГО КОНВЕЙЕРА ДЛЯ RPG С ЦЕЛЮ ЭФФЕКТИВНОГО ПРОТОТИПИРОВАНИЯ

Аннотация. В статье подробно рассматриваются ключевые архитектурные решения, совместные рабочие процессы, модульная практика и стратегии оптимизации производительности. На примере разработки типичной RPG-системы демонстрируются значительные преимущества подхода в скорости создания прототипов и снижении затрат на командную работу, что представляет высоко практичное решение для малых и средних команд.

Ключевые слова: Unity3D, Blender, конвейер разработки игр, RPG, прототипирование, компонентная архитектура.

Для небольших команд разработчиков игр ключевым фактором успеха проекта является возможность быстро проверить концепцию продукта и отточить геймплей в условиях ограниченного бюджета и человеческих ресурсов. Традиционные производственные конвейеры часто страдают от разрозненности инструментов и Совместная работа в сложных процессах, что приводит к снижению производительности, особенно на этапах, связывающих трехмерное моделирование, анимацию и интеграцию в движок.

Для решения этих проблем в данном исследовании предлагается и глубоко анализируется комплексный производственный конвейер, основанный на связке Unity3D (мощный кроссплатформенный игровой движок) и Blender (многофункциональный пакет для создания 3D-графики с открытым исходным кодом). Данное решение не только использует преимущество нулевой стоимости Blender, но и, благодаря тщательно продуманному процессу, глубоко интегрирует оба инструмента, создавая «скоростной путь» от визуальной концепции до интерактивного прототипа. Основная ценность исследования заключается в том, что оно выходит за рамки простого объединения инструментов, предлагая малым командам с точки зрения системной инженерии эффективную, применимую на практике и воспроизводимую парадигму разработки.

Ядром данного конвейера является создание плавного, линейного рабочего процесса без обратных ходов, охватывающего три этапа: от генерации идеи до функциональной реализации.

На этом этапе художники и геймдизайнеры используют базовое скульптурирование меша и плагины для концептуального рисования (такие как Grease Pencil) в Blender для быстрого создания визуальных прототипов. Преимущество заключается в том, что любая концепция может быть непосредственно преобразована в 3D-белую модель в Blender, что закладывает основу для последующего детального моделирования и избегает недопонимания при переходе от 2D-рисунка к 3D-модели.

Это центральный этап конвейера. Все 3D-ассеты, включая персонажей, реквизит сцены, окружение и т.д., создаются в Blender.

- Моделирование и UV-развертка: Использование эффективного набора инструментов моделирования Blender для создания высоко- и низкополигональных моделей и выполнение корректной UV-развертки.

- Процедурная генерация ассетов: Для повторяющихся задач, таких как оформление окружения, инновационно применяется система Геометрических Нодов (Geometry Nodes) Blender. Проектируя нодовые графы, можно быстро генерировать леса, городские постройки, сложный рельеф и т.д., что значительно повышает эффективность и



согласованность построения сцены. Изменение параметров позволяет создавать множество вариаций, обогащая разнообразие игрового мира.

- Риггинг и анимация: Использование мощной и гибкой системы костей и анимационной системы Blender для создания анимаций персонажа (ходьба, бег, прыжки, атаки и т.д.). Его нелинейный редактор анимации позволяет смешивать и редактировать клипы, что удобно для создания сложных анимационных последовательностей.

- Материалы и реальный превью: С помощью рендера реального времени Eevee художники могут получить в Blender эффекты материалов и освещения, близкие к финальным в Unity, что позволяет быстро проводить итерации и корректировки. Это обеспечивает визуальную согласованность ассетов до и после импорта в движок, сокращая время на повторную настройку.

На рисунке 1 представлена общая схема рабочего процесса.



Рисунок 1.

Готовые ассеты экспортируются в форматах FBX или glTF. Оба формата хорошо передают данные о модели, материалах, костях и анимации. В Unity, благодаря предварительной настройке параметров импорта, достигается «бесшовный» импорт ресурсов. Анимационные ассеты напрямую связываются с Контроллерами Аниматора (Animator Controller) Unity, которые управляют поведением персонажа через state machine, обеспечивая плавный переход от анимации в Blender к логике в Unity.



Бесшовный конвейер ассетов предоставляет «снаряды» для эффективной разработки, а система GameObject-Component (Компонент) Unity – это «конвейерная линия», которая собирает эти «снаряды» и вдыхает в них жизнь. Наш конвейер подчеркивает важность компонентного подхода к архитектуре всего RPG-проекта.

На примере RPG-системы, разработанной в исследовании, мы отказались от традиционного монолитного скрипта управления персонажем в пользу разбивки его функциональности на несколько высокосвязных, слабосвязанных независимых компонентов:

- **CharacterCoreComponent:** Отвечает за базовые характеристики и логику: здоровье, ману, перемещение.
- **InventoryComponent:** Независимо управляет системой инвентаря персонажа, обрабатывает добавление, удаление, использование и экипировку предметов. Этот компонент может быть использован любым объектом, нуждающимся в инвентаре: игроком, сундуком, магазином.
- **SkillSystemComponent:** Отвечает за загрузку, откат и применение умений, взаимодействуя с CharacterCoreComponent для нанесения урона или лечения.
- **QuestManager:** Выступая в качестве синглтона, глобально управляет взятием, обновлением прогресса и завершением квестов, уведомляя UI-систему через механизм событий.

Преимущества такой архитектуры очевидны:

1. **Высокая переиспользуемость:** InventoryComponent может быть использован как для игрока, так и для NPC-торговца без дублирования кода.
2. **Слабая связанность и удобство поддержки:** Изменения в боевой логике не затрагивают систему заданий. Каждый модуль имеет четкие обязанности, что облегчает параллельную разработку и отладку.
3. **Быстрые итерации:** Дизайнеры могут быстро создавать новые игровые сущности (например, NPC, который может и сражаться, и давать задания), просто добавляя или настраивая различные компоненты на GameObject прямо в редакторе Unity, что значительно ускоряет создание прототипов.

На рисунке 2 представлена схема компонентной архитектуры.

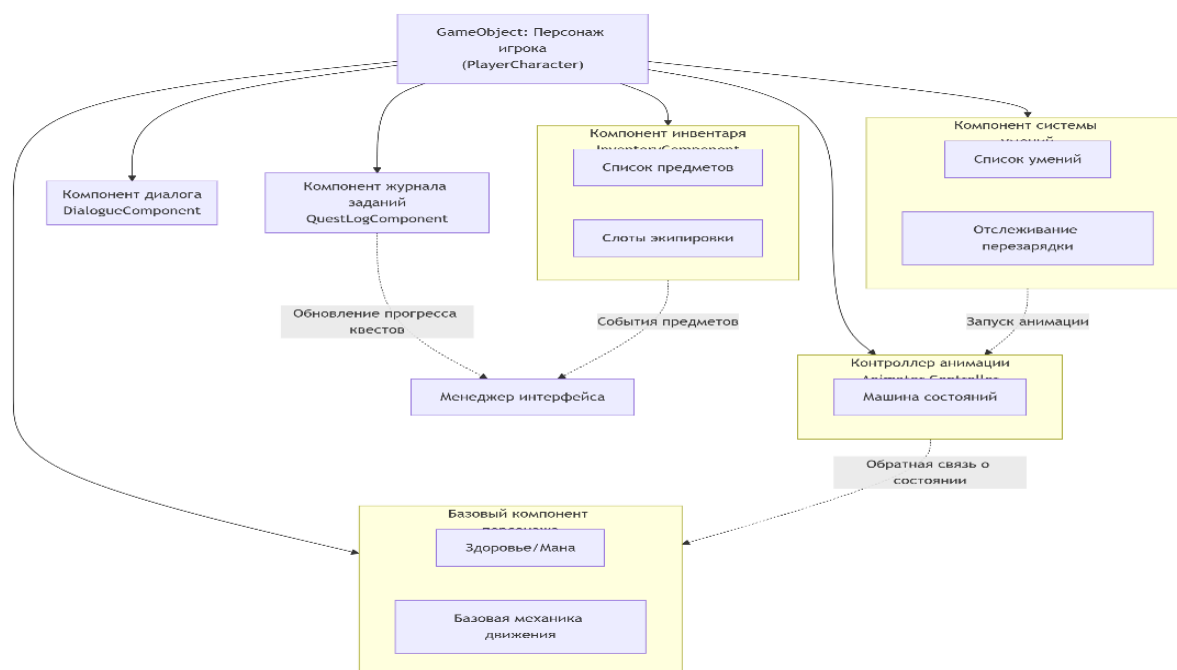


Рисунок 2.



4. Стратегии оптимизации производительности и кроссплатформенной адаптации

Комплексный конвейер учитывает не только эффективность разработки, но и вопросы производительности.

- Оптимизация моделей и текстур: После моделирования в Blender строго выполняются оптимизация топологии и создание уровней детализации (LOD). Для разных целевых платформ (ПК vs. мобильные устройства) устанавливаются различные бюджеты полигонов и стандарты разрешения текстур (например, для мобильных устройств используется формат сжатия ASTC). Конвейер требует, чтобы все ассеты проходили проверку на соответствие стандартам перед экспортом.

- Оптимизация освещения и рендеринга: Используя материалы, настроенные на основе превью в Blender EEVEE, в Unity выбирается подходящий конвейер рендеринга (встроенный или Universal Render Pipeline – URP) в зависимости от платформы. Для мобильных платформ активно применяется запекание карт освещения (lightmapping) вместо динамического освещения для экономии вычислительных ресурсов.

- Оптимизация анимации и кода: В Unity производится оптимизация иерархии состояний в контроллерах аниматора и используется пул объектов (Object Pooling) для управления часто создаваемыми и уничтожаемыми объектами (например, частицами эффектов умений), что обеспечивает плавную работу игры даже на слабых устройствах.

Установив четкие бюджетные ограничения и контрольные точки на каждом из этапов, данный конвейер гарантирует, что конечное игровое приложение будет демонстрировать стабильную производительность на целевых платформах.

Представленное в статье исследование производственного конвейера для RPG, основанного на взаимодействии Unity3D и Blender, демонстрирует, как глубокая интеграция бесшовного процесса «концепция-ассет-интеграция» и компонентной программной архитектуры предоставляет малым и средним командам разработчиков эффективное и экономичное решение, прошедшее практическую проверку.

Список литературы:

1. Воган У. Цифровое моделирование / У. Воган // ДМК Пресс. – 2022.
2. Борромео Н.А. Разработка игр на Unity, 4-е изд. / Н.А. Борромео, С.Х.Г. Гомила // Sprint Book. – 2025.

