

ПРОЦЕСС СОЗДАНИЯ ИГР: ОТ ИДЕИ ДО РЕЛИЗА

*О.В. Федотова², ученица 9 класса,
Научный руководитель: Разумников С.В.^{1а}, к.т.н., доц.
¹Юргинский технологический институт (филиал)
Национального исследовательского Томского политехнического университета
652055, Кемеровская обл., г. Юрга, ул. Ленинградская, 26
²МБОУ «СОШ № 14 имени К.С. Федоровского»
652057, Кемеровская обл., г. Юрга, ул. Волгоградская, 3
E-mail: "razumnikov@tpu.ru*

Аннотация. В статье рассматриваются процесс создания игр, от идеи до релиза, а также специализированные инструменты и программное обеспечение для их разработки.

Ключевые слова: процесс, игры, создание, программирование, жанр, идея.

Abstract: The article discusses the process of creating games, from idea to release, as well as specialized tools and software for their development.

Keywords: process, games, creation, programming, genre, idea.

Введение. В мире видеоигр процесс разработки – это сложный путь, который включает множество этапов и требует участия различных специалистов. От рождения идеи до выхода игры на рынок проходит долгий период, наполненный креативными решениями, техническими вызовами и управленческими задачами. В этой статье мы рассмотрим ключевые этапы процесса создания игр, начиная с зарождения концепции и заканчивая релизом готового продукта.

Этап 1: Идея и концепция

Все начинается с идеи. Это может быть простой замысел сюжета, механики или даже атмосферы, которую хочется передать игроку. На этом этапе важно понять, чем именно будет уникальна игра, какие эмоции она вызовет у игроков и какую нишу займет на рынке. Часто идея рождается благодаря личным интересам разработчиков, вдохновению от других игр или культурным явлениям.

После того как базовая идея сформирована, команда начинает детализировать концепцию:

- Жанр – определяется жанр игры (экшн, ролевая игра, стратегия и так далее).
- Целевая аудитория – анализируется, для кого создается игра: дети, подростки, взрослые игроки?
- Платформы – выбираются платформы, на которых планируется выпуск игры (ПК, консоли, мобильные устройства).

На данном этапе также создаются первые наброски персонажей, мира и ключевых игровых механик. Концепция оформляется в виде документа, который станет основой для дальнейшей работы команды.

Этап 2: Прототипирование

Прототипирование – важный этап, позволяющий проверить жизнеспособность идей и механик. Здесь разработчики создают упрощенную версию игры, чтобы оценить её основные элементы: игровой процесс, интерфейс, управление и баланс. Прототипы могут быть очень простыми – без графики высокого качества и сложных анимаций, но они должны отражать суть будущей игры.

Цель прототипирования – выявить слабые места и внести необходимые изменения до начала полноценной разработки. Этот этап помогает сэкономить ресурсы и избежать ошибок на поздних стадиях проекта.

Этап 3: Дизайн и документация

Когда концепция утверждена и прототип доказал свою работоспособность, наступает время детальной проработки дизайна. В этот момент создаются подробные документы для игры:

- Дизайн-документ – содержит полное описание игрового процесса, правил, уровней, механик и других элементов.
- Техническое задание – документ, определяющий технические требования к игре, включая используемые технологии, платформу и инструменты разработки.
- Арт-бук – сборник визуальных материалов, таких как концепт-арты, скетчи и модели персонажей, которые будут использоваться художниками при создании графики.

Документация играет ключевую роль в организации работы команды и позволяет всем участникам понимать свои задачи и цели.

Этап 4: Разработка

Этот этап является самым продолжительным и трудоемким. Он включает несколько подэтапов:

Программирование

Программисты начинают писать код, реализуя игровые механики, искусственный интеллект, физику и другие технические аспекты. Важно использовать современные инструменты и фреймворки, такие как Unity, Unreal Engine или собственные движки, чтобы обеспечить производительность и игры.

Графика и анимация

Художники и аниматоры работают над созданием визуального контента: моделей персонажей, окружения, эффектов и интерфейсов. Современные игры требуют высокой детализации и реалистичности, поэтому этот этап часто занимает много времени и ресурсов.

Звук и музыка

Звукорежиссеры записывают звуковые эффекты и музыку, которые помогут создать нужную атмосферу и усилить эмоциональный отклик игрока. Музыка должна соответствовать стилю игры и поддерживать игровую динамику.

Тестирование

Параллельно с разработкой идет тестирование. Тестировщики проверяют игру на наличие багов, ошибок и несоответствий дизайн-документу. Они также оценивают удобство управления и общую играбельность. Найденные проблемы фиксируются и отправляются разработчикам для исправления.

Этап 5: Альфа-версия и бета-тестирование

Альфа-версия – это первая версия игры, которая уже имеет большинство основных функций, но еще не полностью завершена. Она используется для внутреннего тестирования командой разработчиков и первыми пользователями (если проводится закрытое альфа-тестирование). Основная цель этого этапа – найти и устранить критические ошибки и недочеты.

Бета-версия – это почти готовая игра, доступная для публичного тестирования. Бета-тестирование позволяет получить обратную связь от широкой аудитории и улучшить игру перед финальным релизом. На основе отзывов пользователей вносятся последние правки и улучшения.

Этап 6: Финальная сборка и релиз

Финальная сборка – это окончательная версия игры, готовая к выходу на рынок. Все баги устранены, контент завершён, а игра оптимизирована для выбранной платформы. Перед выпуском проводится последний раунд тестирования, чтобы убедиться в отсутствии серьезных проблем.

Релиз включает в себя публикацию игры на цифровых платформах (Steam, PlayStation Store, App Store и др.), маркетинговую кампанию и поддержку после выпуска. Команда продолжает следить за игрой, выпускать обновления и патчи, а также взаимодействовать с сообществом игроков.

Для разработки игр используются различные специализированные инструменты и программное обеспечение. Среди них можно выделить следующие категории:

1. Игровые движки: Это наборы типичных компонентов, необходимых для разработки игр, включающие визуальные инструменты и интегрированные среды разработки. Примером такого инструмента является CryEngine 2, который предоставляет полный функционал для создания игр, включая графический движок, систему анимации персонажей, физический движок, звуковой движок, искусственный интеллект и многое другое.

2. Редакторы уровней: Специальные программы, предназначенные для создания уровней, задач или миссий для компьютерных видеоигр. Они могут быть общедоступными коммерческими программами 3D или 2D проектирования, либо разработанными редакторами для конкретных игр.

3. Инструменты разработки игр: Вспомогательное специализированное программное обеспечение, которое сопровождает разработку игры и используется дизайнерами и сценаристами. Такие инструменты могут включать прототипы и программы для преобразования активов, а также редакторы уровней, которые иногда поставляются вместе с игрой.

4. Языки программирования: Выбор языка разработки зависит от знаний программистов, целевых платформ и требований к проекту. Популярные языки для разработки игр включают C++, JavaScript, Python и другие.

5. Средства анализа производительности: Включают инструменты для мониторинга и оптимизации производительности игры в реальном времени, создания отчетов об использовании памяти и выполнения автоматизированных тестов.

Заключение. Создание игр – это сложный и многогранный процесс, требующий координации усилий множества специалистов. Каждый этап важен и влияет на конечный результат.

Успех игры зависит не только от хорошей идеи, но и от тщательной реализации всех аспектов: дизайна, программирования, графики, звука и тестирования. Благодаря слаженной работе команды и вниманию к деталям, на свет появляются шедевры, которые захватывают сердца миллионов игроков по всему миру.

Список использованных источников:

1. Казакова Н.Ю. Этап тестирования игрового проекта в гейм-дизайне как виде проектной деятельности / Н.Ю. Казакова // Академическая наука-проблемы и достижения. – 2016. – С. 6–10.
2. Жукова М.С. Особенности создания десктопных 2D игр в жанре test / М.С. Жукова // Наука России 3. – 2025. – С. 17.
3. Кадыров П.Р. Оптимизация процесса создания игровых ассетов при помощи использования нейронных сетей / П.Р. Кадыров // Вестник науки. – 2024. – Т. 5. – № 12 (81). – С. 688–697.

ОПТИМИЗАЦИЯ ПРОИЗВОДИТЕЛЬНОСТИ WebGL-ПРИЛОЖЕНИЙ С ИСПОЛЬЗОВАНИЕМ МЕТОДОВ ЛИНЕЙНОЙ АЛГЕБРЫ

И.С. Потапов^a, студент гр. 17В41,

Научные руководители: Воробьев А.В., к.т.н., доц., Гиль Л.Б., к.пед.н., доц.

Юргинский технологический институт (филиал)

Национального исследовательского Томского политехнического университета

652055, Кемеровская обл., г. Юрга, ул. Ленинградская, 26

E-mail: ^adaisy40@inbox.ru

Аннотация: В данной статье рассматриваются методы применения линейной алгебры для улучшения WebGL-приложений. К рассматриваемым методам относятся: снижение количества матричных операций, совершенствование геометрических преобразований, а также использование разреженных матриц в 3D-моделировании.

Ключевые слова: WebGL, линейная алгебра, матрицы, разреженные матрицы, геометрические преобразования, 3D-моделирование.

Abstract: This article explores methods for applying linear algebra to enhance WebGL applications. The methods considered include the number of matrix operations, improving geometric transformations, and utilizing sparse matrices in 3D modeling.

Keywords: WebGL, linear algebra, matrices, sparse matrices, geometric transformations, 3D modeling.

WebGL – API для языка программирования JavaScript, оно предназначено для отображения 2D и 3D графики[1]. Его популярность объясняется универсальной работой на любой платформе и возможностью создания трёхмерной графики прямо в браузере без использования дополнительных расширений [2]. 3D-графика сама по себе является очень трудоёмкой задачей, которая требует сложных вычислений, что не каждое устройство способно осилить.

Большинство операций WebGL основываются на применении принципов линейной алгебры. Как и в языках программирования, в математике в решении задач можно пользоваться разными методами, одни методы могут быть рациональнее других. Это можно отнести и к сфере 3D-моделирования в браузере, в которой от выбранного подхода зависит количество вычислений, требуемых устройству для отображения графики.

Вычисления можно свести к минимуму, грамотно выбирая подходы в линейной алгебре и комбинируя это с WebGL.

К основным понятиям линейной алгебры, применяемым в WebGL, относятся матрицы, векторы и преобразования. Назначение матриц – это изменение масштаба, вращение и проекция объекта.

Конвейер рендеринга WebGL использует матрицы в шейдерах вершин, чтобы превратить координаты вершин из пространства объекта в пространство отсечения.

Все вершины умножаются на модально-видовую матрицу и проекционную, таким образом от этих матричных операций прямо зависит производительность рендеринга.