

Литература

1. Lazarev, L. A. Stress-deformed state of the proximal branch of the cell of fibrous dysplasia under conditions of osteosynthesis by different types of clamps / I. A. Lazarev [et al.] // Trauma. – 2015. – Vol. 16. – Pp. 49–56.

2. Trajanovic, M. Application of computer models of Mitcovic selfdynabizable internal fixator in rehabilitation of femur traumas / M. Trajanovic [et al.] // Facta Universitatis. – 2010. – Vol. 8. – P. 27–38.

И. В. Гаращук

(ГГТУ имени П. О. Сухого, Гомель)

Науч. рук. **И. Л. Стефановский**, ст. преподаватель

ИГРОВОЕ 2D ПРИЛОЖЕНИЕ «НАСЛЕДИЕ БИТВЫ» В ЖАНРЕ «ФАЙТИНГ» ДЛЯ ДВУХ ИГРОКОВ С ИСПОЛЬЗОВАНИЕМ ИГРОВОГО ДВИЖКА UNITY

Введение. Современные технологии разработки игр позволяют создавать качественные проекты даже небольшим командам. Unity как один из ведущих игровых движков предоставляет все необходимые инструменты для реализации 2D-игр различной сложности. Особый интерес представляет разработка файтингов – жанра, где важную роль играет точность управления, баланс персонажей и отзывчивость игровых механик.

Проект «Наследие битвы» представляет собой классический 2D-файтинг. Основной акцент при разработке был сделан на создание сбалансированной боевой системы с двумя типами атак и механизмом защиты, а также реализацию платформенных элементов, добавляющих глубину игровому процессу.

Результаты и обсуждение. Разработка игрового приложения «Наследие битвы» потребовала комплексного подхода к реализации ключевых игровых механик. Центральным элементом проекта стала боевая система, которая разрабатывалась с учетом необходимости обеспечения баланса между персонажами и создания увлекательного игрового процесса. Основой боевой механики стали два типа атак – быстрая и мощная, каждый из которых имеет свои уникальные характеристики. Быстрые атаки позволяют игрокам наносить серии ударов с минимальной задержкой между ними, но обладают относительно небольшим уроном. В отличие от них, мощные атаки требуют больше времени для выполнения и имеют заметную задержку перед ударом, но компенсируют это значительным уроном.

Реализация системы ударов потребовала тщательной настройки анимационных клипов и соответствующих им коллайдеров, которые активируются только в определенные моменты выполнения анимации. Это обеспечивает точное соответствие между визуальным представлением атаки и ее фактическим воздействием в игровом мире. Мультиплатформенная разработка на C# подчеркивается важность синхронизации визуальных и игровых элементов для создания убедительного игрового опыта [1]. Особое внимание было уделено системе обратной связи – каждый успешный удар сопровождается звуковыми эффектами и визуальными индикаторами, что значительно усиливает ощущение воздействия.

Важнейшим компонентом игрового процесса стала механика защиты, реализованная через систему блокирования. При активации щита персонаж не получает входящий урон, но при этом теряет возможность атаковать и перемещаться с обычной скоростью. Баланс этой механики достигается за счет введения ограниченного ресурса щита, который постепенно расходуется при блокировании атак и восстанавливается, когда игрок не использует защиту. Техническая реализация системы защиты потребовала создания

отдельного слоя коллайдеров, которые взаимодействуют с атакующими ударами по особым правилам. Визуально состояние защиты подчеркивается изменением цвета персонажа, появлением защитного поля и специальными эффектами частиц при успешном блокировании.

Платформенная составляющая игры была реализована с использованием физического движка Unity. Каждый персонаж обладает способностью к одиночным прыжкам. Правильная настройка параметров прыжка является ключевым фактором для комфортного управления в играх [2]. Особенностью реализации стало введение системы «воздушного контроля», позволяющей игрокам корректировать траекторию движения в прыжке, но с определенными ограничениями, сохраняющими баланс. Для обработки взаимодействия с платформами используется система слоев физики, включающая специальные платформы, через которые персонажи могут проваливаться вниз. Реализация этой механики потребовала тщательной настройки параметров физики и создания специальных алгоритмов обработки столкновений, учитывающих направление движения персонажа и его текущее состояние. Дополнительной сложностью стало обеспечение плавности перемещения между платформами разной высоты, для чего был разработан механизм плавной коррекции позиции персонажа при приземлении.

Визуальная составляющая игры была построена на профессиональной системе анимации, созданной с использованием Unity Animator. Каждое действие персонажа имеет уникальные анимационные клипы с плавными переходами между состояниями. Особое внимание было уделено синхронизации анимаций с игровыми событиями – например, скорость проигрывания анимации прыжка динамически изменяется в зависимости от высоты прыжка. Система частиц используется для создания эффектов ударов, блокирования и других игровых событий, значительно усиливая визуальное воздействие. Динамическая камера автоматически отслеживает положение обоих игроков, обеспечивая оптимальный обзор игрового пространства без потери важных деталей. Интерфейс игры выполнен в минималистичном стиле и включает только необходимые элементы: индикаторы здоровья, оставшихся жизней, что позволяет игрокам полностью сосредоточиться на бое.

Технической основой проекта стала модульная архитектура, где каждый компонент игры реализован как независимый модуль. Такой подход значительно упростил процесс тестирования и балансировки игрового процесса. Все параметры персонажей и игровых механик вынесены в настраиваемые ассеты, что позволяет изменять баланс игры без модификации исходного кода. Для обработки ввода используется система событий, обеспечивающая точную реакцию на действия игроков. Оптимизация производительности достигается за счет кэширования ресурсов, батчинга спрайтов и эффективного управления памятью, что особенно важно для обеспечения стабильной частоты кадров в интенсивных игровых сценах.

Заключение. Разработанное игровое приложение демонстрирует эффективность использования Unity для создания качественных 2D-файтингов. Реализованные механики обеспечивают глубокий и сбалансированный игровой процесс, а модульная архитектура проекта позволяет легко расширять его новыми персонажами и уровнями. Перспективы развития проекта включают добавление новых игровых персонажей, ударов и различных арен.

Литература

1. Хокинг, Дж. Unity в действии: Мультиплатформенная разработка на C# / Дж. Хокинг. – М. : ДМК Пресс, 2023. – 480 с.
2. Грегори, Д. Игровая физика. Алгоритмы и реализация / Д. Грегори. – СПб. : Питер, 2020. – 896 с.