МИНОБРНАУКИ РОССИИ
ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Колледж информационных и креативных технологий

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

МДК.06.02. Продвинутая разработка игровых систем Unreal

программы подготовки специалистов среднего звена
09.02.10 Разработка компьютерных игр, дополненной и виртуальной реальности

Форма обучения: очная

Владивосток 2026

Рабочая программа учебной дисциплины МДК.06.02. Продвинутая разработка игровых систем Unreal разработана в соответствии с требованиями Федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.10 Разработка компьютерных игр, дополненной и виртуальной реальности, утвержденного приказом Минобрнауки России от 25 июня 2024, № 441, примерной образовательной программой.

Разработчик(и):О.Ю. Рудзейт, преподаватель

Рассмотрено и одобрено на заседании цикловой методической комиссии

Протокол № _9_ от «_20_» ___мая____2026_ г.

Председатель ПЦК _____ Н.В. Кравцова

подпись

СОДЕРЖАНИЕ

1.	ОБЩАЯ ХАРАКТЕРИСТИКА ПРИМЕРНОЙ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ	3
2.	СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ	5
3.	УСЛОВИЯ РЕАЛИЗАЦИИ УЧЕБНОЙ ДИСЦИПЛИНЫ	10
4.	КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ	11

1. ОБЩАЯ ХАРАКТЕРИСТИКА ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

1.1. Место дисциплины в структуре основной образовательной программы:

Учебная дисциплина МДК.06.02 «Продвинутая разработка игровых систем Unreal» является обязательной частью технического цикла основной образовательной программы в соответствии с ФГОС СПО по специальности 09.02.10 Разработка компьютерных игр, дополненной и виртуальной реальности

1.2. Цель и планируемые результаты освоения дисциплины:

В рамках программы учебной дисциплины обучающимися осваиваются умения и знания

Код компетенции	Умения	Знания
ПК -6.1 *(из ПООП соответствующей профессии)*	Уметь: Использовать популярные платформы для сборки, настройки и развертывания контента; использовать основные платформы сборки проектов на Unreal Engine 5 (Windows, macOS, Linux, Android, iOS, консоли); применять настройки проекта (Project Settings) для целевых платформ: разрешение, соотношение сторон, управление памятью, API рендеринга (DirectX 12, Vulkan, Metal); работать с профилями сборки (Development, Shipping, Debug) и их влияние на производительность и отладку; применять инструменты упаковки контента (Content Browser, Asset Manager, Chunking) для оптимизации дистрибутива; работать с системами контроля версий для командной разработки и слияния контента	Знать: Настраивать проект под сборку для выбранной платформы (смена целевой платформы, конфигурация графических настроек). Проводить сборку (Package Project) для Windows, Android и iOS с использованием Device Manager. Оптимизировать размер и структуру пакетов контента (разделение на чанки, сжатие текстур, выгрузка ненужных ассетов). Развертывать собранное приложение на тестовых устройствах.
ПК-6.3 *(из ПООП соответствующей профессии*	Уметь: Разрабатывать механику игрового процесса; проектировать и реализовывать базовые и сложные игровые механики с использованием Blueprints и C++; настраивать систему ввода для поддержки клавиатуры, мыши, геймпадов и тач-интерфейсов; создавать интерактивные системы: инвентарь, квесты, диалоги, боевые механики, физику взаимодействия; интегрировать анимации и звуковые эффекты в игровые механики через Animation Notify и Audio Components;	Знать: Архитектуру игрового фреймворка Unreal Engine 5: GameMode, GameState, PlayerController, Pawn, Character; принципы компонентного подхода и композиции объектов в UE5; систему ввода Enhanced Input System и её настройку под разные устройства; механизмы взаимодействия между актерами: интерфейсы, делегаты, события, ноды Communication; основы проектирования игровых циклов, состояний и конечных автоматов (State Machines);

	проводить итеративное тестирование и балансировку механик на основе фидбека и метрик; оптимизировать логику геймплея для снижения нагрузки на CPU (минимизация Tick, использование таймеров и событий); документировать архитектуру механик для передачи команде и поддержки кода.	методы балансировки игровой механики на этапе прототипирования; паттерны проектирования, применимые в геймдеве (Observer, Command, Strategy, Singleton).
ПК-6.4 *(из ПООП соответствующей профессии*	Уметь: Программировать игровую графику и специальные эффекты; создавать и настраивать сложные материалы и шейдеры для реалистичной и стилизованной графики; разрабатывать визуальные эффекты (VFX) в Niagara: взрывы, магия, погода, интерфейсные эффекты; оптимизировать графический контент: настраивать LOD, атласы текстур, сжатие, инстансинг; настраивать динамическое освещение и тени с учётом производительности целевой платформы; проводить профилирование графики (Stat GPU, Unreal Insights) и устранять узкие места рендеринга.	Знать: Архитектуру рендеринга Unreal Engine 5: Deferred Shading, Nanite, Lumen, Virtual Shadow Maps; систему частиц и визуальных эффектов: Niagara, Cascade (legacy), GPU Particles; методы оптимизации графики: LOD, HLOD, culling, batching, texture streaming; работу с пост-процессингом: Volume, Post Process Material, Color Grading, Bloom, DOF; принципы работы с освещением: Static/Dynamic/Stationary lights, Lightmass, Light Functions; Интеграцию визуальных скриптов и C++ кода для управления графическими системами.
ПК-6.5 *(из ПООП соответствующей профессии*	Уметь: Разрабатывать системы игрового баланса; проектировать и настраивать формулы баланса с использованием Data-Driven подхода (Data Assets, Tables); реализовывать системы отслеживания игровых метрик и аналитики внутри проекта; проводить симуляции и итеративную настройку параметров баланса на основе тестовых данных; создавать инструменты для дизайнеров: редакторы баланса в редакторе, hot-reload конфигураций; балансировать сложность, прогресс и награды с учётом целевой аудитории и платформы; интегрировать системы динамической сложности (Dynamic Difficulty Adjustment) при необходимости.	Знать: Теорию игрового баланса: кривые прогрессии, экономику, вероятность, риск/вознаграждение; методы математического моделирования баланса: формулы урона, опыта, стоимости, времени; инструменты анализа данных в UE5: Stats, Analytics, Replay System, Custom HUD-метрики; принципы проектирования внутриигровой экономики: валюты, ресурсы, магазины, награды; Методы A/B-тестирования и сбора обратной связи от игроков на ранних этапах; Основы поведенческой психологии игрока и её влияние на баланс; Инструменты визуализации данных: таблицы (Data Assets), графики.

2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

2.1. Объем учебной дисциплины и виды учебной работы

Вид учебной работы	Объем часов
Объем образовательной программы учебной дисциплины	306
в том числе:	
– теоретическое обучение	118
– практические занятия *(если предусмотрено)*	118
– лабораторные занятия *(если предусмотрено)*	0
– курсовая работа (проект) *(если предусмотрено)*	0
– самостоятельная работа	64
– консультации	2
– промежуточная аттестация – *(форма промежуточной аттестации)*	4

2.2. Тематический план и содержание учебной дисциплины

Наименование разделов и тем	Содержание учебного материала и формы организации деятельности обучающихся	Объем в часах	Коды компетенций, формированию которых способствует элемент программы
1	2	3	4
Раздел 1. Основы архитектуры Unreal Engine 5			
Тема 1.1. Введение в архитектуру UE5. Система модулей	**Архитектура движка Unreal Engine 5. Модульная структура. Система сборки. Core и Engine модули**	5	ПК.6.1
	Практическая работа №1.1: Настройка рабочего окружения. Анализ структуры проекта	5	
	Самостоятельная работа обучающихся Изучение документации по модульной системе UE5. Подготовка отчета о структуре движка	3	
Тема 1.2. Система типов данных и рефлексия	**Система типов Unreal. UPROPERTY, UFUNCTION. Система рефлексии. Сборка мусора**	5	ПК.6.3
	Практическая работа №1.2: Создание кастомных классов с использованием макросов UE5	5	
	Самостоятельная работа обучающихся Разработка тестовых классов с различными спецификаторами свойств	2	
Тема 1.3. Архитектура игрового фреймворка	**GameMode, GameState, PlayerController, Pawn, Character. Жизненный цикл объектов**	6	ПК.6.3
	Практическая работа №1.3: Реализация кастомного GameMode и PlayerController	6	
	Самостоятельная работа обучающихся Создание диаграммы взаимодействия компонентов фреймворка	3	
	Компонентная архитектура. Компоненты актеров. Сценные компоненты	5	

Тема 1.4. Система компонентов и композиция	Практическая работа №1.4: Разработка кастомных компонентов для повторного использования	5	ПК.6.3
	Самостоятельная работа обучающихся **Создание библиотеки переиспользуемых компонентов**	2	
Тема 1.5 Игровые события и интерфейсы	**Интерфейсы Blueprint, события, диспетчеры.**	4	ПК.6.3
	Практическая работа №1.5: Создание двери, открывающейся по взаимодействию.	4	
	Самостоятельная работа обучающихся Разработка простого квеста с активацией по событию	8	
Раздел 2. Система рендеринга и графическая архитектура			
Тема 2.1. Архитектура рендеринга UE5	**Отложенное затенение. Конвейер рендеринга. RHI**	7	ПК.6.4
	Практическая работа №2.1: Анализ графического конвейера. Stat команды	7	
	Самостоятельная работа обучающихся Изучение документации по RHI. Сравнение API (DX11, DX12, Vulkan)	4	
Тема 2.2. Nanite и виртуальная геометрия	**Принцип работы Nanite. Виртуализированная геометрия. Иерархия кластеров**	6	ПК.6.4
	Практическая работа №2.2: Настройка Nanite для высокополигональных моделей	6	
	Самостоятельная работа обучающихся Оптимизация сцены с использованием Nanite. Сравнительный анализ FPS	3	
Тема 2.3. Lumen и глобальное освещение	**Динамическое глобальное освещение. Трассировка лучей против программной трассировки лучей**	6	ПК.6.4
	Практическая работа №2.3: Настройка Lumen для различных типов сцен	6	
	Самостоятельная работа обучающихся Исследование влияния настроек Lumen на производительность	3	

Тема 2.4. Система материалов и шейдеров	**Граф материалов. Компиляция шейдеров. Инстансы материалов. Функции материалов**	7	ПК.6.4
	Практическая работа №2.4: Создание оптимизированных материалов с использованием инстансов	7	
	Самостоятельная работа обучающихся **Разработка библиотеки материалов. Документирование**	3	
Тема 2.5 Пост-обработка	**Post Process Volume: цветокоррекция, глубина резкости, bloom**	2	ПК.6.4
	Пр.р №2.5. Настройка пост-эффектов (кино-фильтры).	2	
	Самостоятельная работа обучающихся Эксперимент: влияние bloom на восприятие сцены.	7	
Раздел 3. Системы оптимизации производительности			
Тема 3.1 Профилирование и анализ производительности	**Unreal Insights. Stat команды. GPU Visualizer. Профайлер памяти**	7	ПК.6.1, ПК.6.4
	Практическая работа №3.1: Комплексный анализ производительности проекта	7	
	Самостоятельная работа обучающихся Создание отчета о производительности с рекомендациями по оптимизации	3	
Тема 3.2. Оптимизация CPU и игровых систем	**Игровой поток против потока рендеринга. Асинхронные задачи. Параллелизация. Data-driven баланс**	7	ПК.6.3, ПК.6.5
	Практическая работа №3.2: Оптимизация игровой логики. Использование async nodes	7	
	Самостоятельная работа обучающихся Рефакторинг кода. Настройка таблиц баланса (Data Assets)	3	
Тема 3.3. Оптимизация GPU и памяти	**Стриминг текстур. Виртуальные текстуры. Пулы памяти. Управление ресурсами**	7	ПК.6.4
	Практическая работа №3.3: Настройка стриминга текстур. Анализ VRAM	7	
	Самостоятельная работа обучающихся Оптимизация использования памяти в сцене	3	

Раздел 4 Система контента и управление ассетами			
Тема 4.1 Asset pipeline и импорт	**Реестр ассетов. Конвейер импорта. Система фабрик. Массивные данные**	5	ПК.6.1
	Практическая работа №4.1: Настройка автоматического импорта ассетов	5	
	Самостоятельная работа обучающихся **Создание гайдлайнов по импорту контента для команды**	2	
Тема 4.2 Управление зависимостями и версиями	**Интеграция с системой контроля версий. Сравнение ассетов. Редиректоры. Управление пакетами**	5	ПК.6.1
	Практическая работа №4.2: Работа с GitUniverse в UE5. Merge конфликты	5	
	Самостоятельная работа обучающихся **Настройка CI/CD pipeline для проекта UE5**	2	
Тема 4.3 Оптимизация контента	**LOD (Уровень детализации). HLOD. Импостеры. Атласирование текстур**	5	ПК.6.1, ПК.6.4
	Практическая работа №4.3: Создание LOD цепочек. Настройка HLOD системы	5	
	Самостоятельная работа обучающихся **Оптимизация существующей сцены с применением всех методов**	2	
Раздел 5 Платформенная специфика и развертывание проекта			
Тема 5.1 Кросс-платформенная разработка	**Абстракция платформ. Конфигурация сборки. Целевые платформы**	5	ПК.6.1
	Практическая работа №5.1: Настройка сборки для Windows, Android, iOS	5	
	Самостоятельная работа обучающихся Сравнительный анализ требований платформ	2	
Тема 5.2 Упаковка и развертывание приложений	**Процесс приготовления (Cooking). Настройки упаковки. Специфичные для платформы оптимизации**	5	ПК.6.1
	Практическая работа №5.2: Сборка и упаковка финального билда	5	
	Самостоятельная работа обучающихся Создание чек-листа развертывания проекта для разных платформ	2	

Тема 5.3 Оптимизация под мобильные платформы	**Мобильный рендеринг. ES3.1 против Vulkan. Ограничения мобильных платформ**	5	ПК.6.1
	Практическая работа №5.3: Адаптация проекта под мобильные устройства	5	
	Самостоятельная работа обучающихся Тестирование и оптимизация под целевое мобильное устройство	2	
Раздел 6 Продвинутые темы архитектуры			
Тема 6.1 Система анимации и Character FX	**Animation Blueprint. Control Rig. Motion Warping. Ретаргетинг**	8	ПК.6.4, ПК 6.5
	Практическая работа №6.1: Создание сложной системы анимации персонажа	8	
	Самостоятельная работа обучающихся Оптимизация Animation Blueprint для производительности	3	
Тема 6.2 Niagara и система частиц	**GPU частицы. Архитектура Niagara. Пользовательские HLSL узлы**	6	ПК.6.4, ПК 6.5
	Практическая работа №6.2: Создание оптимизированных VFX эффектов	6	
	Самостоятельная работа обучающихся Сравнение систем частиц по производительности	2	
Консультации		2	
Промежуточная аттестация (форма промежуточной аттестации)		4	
Всего:		306	

3. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

3.1. Для реализации программы учебной дисциплины должны быть предусмотрены следующие специальные помещения:

Кабинет «Продвинутая разработка игровых систем Unreal», оснащённый:

- *оборудованием:*

посадочные места по количеству обучающихся;

рабочее место преподавателя;

- *техническими средствами обучения:*

компьютер (ноутбук) с лицензионным программным обеспечением для преподавателя;

компьютер (ноутбук) с лицензионным программным обеспечением на каждое посадочное место по количеству обучающихся;

мультимедийный проектор;

мультимедийный экран;

информационно-коммуникативные средства;

экранно-звуковые пособия

3.2. Информационное обеспечение реализации программы

Для реализации программы библиотечный фонд образовательной организации должен иметь печатные и/или электронные образовательные и информационные ресурсы, для использования в образовательном процессе. При формировании библиотечного фонда образовательной организацией выбирается не менее одного издания из перечисленных ниже печатных изданий и (или) электронных изданий в качестве основного, при этом список может быть дополнен новыми изданиями.

3.2.1. Основные печатные издания

1. Иванов, А. В. C++ и Unreal Engine 5: разработка игровых механик / А. В. Иванов. – М. : Технобук, 2024. – 280 с.

2. Воронцов, Д. А. Визуальное программирование в Unreal Engine 5: Blueprints : учебное пособие / Д. А. Воронцов. – СПб. : БХВ-Петербург, 2024. – 400 с.

3. Гриффин, Дж. Unreal Engine 5. Разработка игр / Дж. Гриффин ; пер. с англ. А. Н. Иванова. – М. : ДМК Пресс, 2023. – 550 с.

3.2.2. Электронные издания

1. Epic Games. Unreal Engine 5 Documentation: [сайт]. – URL: unrealengine.com (дата обращения: 01.04.2026). – Текст : электронный.

4 МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ИЗУЧЕНИЮ И РЕАЛИЗАЦИИ МЕЖДИСЦИПЛИНАРНОГО КУРСА

Начиная изучение дисциплины, студенту необходимо:

- ознакомиться с программой, изучить список рекомендуемой литературы;

- внимательно разобраться в структуре курса, в системе распределения учебного материала по видам занятий, формам контроля, чтобы иметь представление о курсе в целом.

Самостоятельная работа студентов (СРС) — это деятельность учащихся, которую они совершают без непосредственной помощи и указаний преподавателя, руководствуясь сформировавшимися ранее представлениями о порядке и правильности выполнения операций. Цель СРС в процессе обучения заключается, как в усвоении знаний, так и в формировании умений и навыков по их использованию в новых условиях на новом учебном

материале. Самостоятельная работа призвана обеспечивать возможность осуществления студентами самостоятельной познавательной деятельности в обучении, и является видом учебного труда, способствующего формированию у студентов самостоятельности. Кроме того, для расширения и углубления знаний по дисциплине целесообразно использовать: публикации в тематических журналах; полнотекстовые базы данных библиотеки; имеющиеся в библиотеках вуза. Успешное освоение дисциплины предполагает активную работу студентов на всех занятиях аудиторной формы: лекций и практических занятий, выполнение аттестационный мероприятий, эффективную самостоятельную работу.

4.1 Методические рекомендации обучающимся по обеспечению самостоятельной работы

Самостоятельная работа проводится с целью: систематизации, закрепления, углубления и расширения теоретических знаний и практических умений, приобретаемых студентами в ходе аудиторных занятий; формирования умений использовать специальную литературу; развития познавательных способностей и активности обучающихся; формирования самостоятельности мышления, способности к саморазвитию, самосовершенствованию и самореализации; развития исследовательских умений.

Для лучшего освоения материала и систематизации знаний по дисциплине необходимо отвечать на вопросы для самоконтроля. Во время самостоятельной проработки лекционного материала особое внимание следует уделять возникшим вопросам, непонятным терминам, спорным точкам зрения. В случае необходимости обращаться к преподавателю за консультацией. Студент должен четко уяснить, что именно с лекции начинается его подготовка к практическому занятию. Вместе с тем, лекция лишь организует мыслительную деятельность, но не обеспечивает глубину усвоения программного материала.

При подготовке к практическому занятию особое внимание необходимо обратить на содержание основных положений и выводов, объяснение явлений и фактов, уяснение практического приложения рассматриваемых теоретических вопросов. В процессе этой работы студент должен стремиться понять и запомнить основные положения рассматриваемого материала, примеры, поясняющие его, а также разобраться в иллюстративном материале. В процессе подготовки рекомендуется взаимное обсуждение материала, во время которого закрепляются знания, а также приобретается практика в изложении и разъяснении полученных знаний, развивается речь.

При необходимости следует обращаться за консультацией к преподавателю. Перед консультацией, необходимо хорошо продумать вопросы, которые требуют разъяснения.

В процессе изучения дисциплины студенту необходимо ориентироваться на самостоятельную проработку лекционного материала, подготовку к практическим занятиям, выполнение тестов, кейсовых заданий, самостоятельное изучение некоторых разделов курса. Для проведения занятий лекционного типа используются учебно-наглядные пособия в форме презентационных материалов, обеспечивающих тематические иллюстрации, соответствующие темам лекций.

4.2 Особенности организации обучения для лиц с ограниченными возможностями здоровья и инвалидов

При необходимости обучающимся из числа лиц с ограниченными возможностями здоровья и инвалидов (по заявлению обучающегося) предоставляется учебная информация в доступных формах с учетом их индивидуальных психофизических особенностей:

для лиц с нарушениями зрения: в печатной форме увеличенным шрифтом; в форме электронного документа; индивидуальные консультации с привлечением тифлосурдопереводчика; индивидуальные задания, консультации и др.

для лиц с нарушениями слуха: в печатной форме; в форме электронного документа; индивидуальные консультации с привлечением сурдопереводчика; индивидуальные задания, консультации и др.

для лиц с нарушениями опорно-двигательного аппарата: в печатной форме; в форме электронного документа; индивидуальные задания, консультации и др.

5 КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ КУРСА

Код ПК	**Критерии оценки результата (показатели освоенности компетенций)**	**Формы контроля и методы оценки**
ПК.6.1	Оценка "отлично": алгоритмы разработаны в полном соответствии с ТЗ, оптимизированы и понятны. Оценка "хорошо": алгоритмы разработаны в соответствии с ТЗ, но допускают незначительные отклонения. Оценка "удовлетворительно": алгоритмы разработаны с частичным соответствием ТЗ.	Проверка оформленного кода, демонстрация работы приложения на различных платформах.
ПК.6.3	Оценка "отлично": программные модули разработаны в полном соответствии с ТЗ, тесты проходят успешно. Оценка "хорошо": программные модули разработаны с минимальными несоответствиями, тесты в целом успешны. Оценка "удовлетворительно": программные модули разработаны с существенными доработками.	Проверка оформленного кода и логика проекта на различных устройствах.
ПК.6.4	Оценка "отлично": код полностью оформлен в соответствии с требованиями, включая комментарии и стиль кода. Оценка "хорошо": код оформлен в соответствии с требованиями, допускаются мелкие недочёты. Оценка "удовлетворительно": код оформлен частично в соответствии с требованиями.	Проверка оформленного кода, соблюдения стиля и соответствия установленным требованиям.
ПК.6.5	Оценка "отлично": игровые элементы хорошо оптимизированы, разработанные математические методы балансировки вычисляют результаты с допустимыми отклонениями.	Демонстрация работы математических алгоритмов, механик

	Оценка "хорошо": наблюдаются небольшие проблемы с работоспособностью окружения, которые несильно влияют на работу проекта. Оценка "удовлетворительно":созданная система работает со сбоями, наблюдаются ошибки, влияющие на игровое окружение.	работы с игровым окружением

4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

4.1. Описание показателей и критериев оценки компетенций

Результаты обучения	*Критерии оценки*	*Методы оценки*
Перечень знаний, осваиваемых в рамках дисциплины		
Знать: – Популярные платформы сборки проектов на Unreal Engine 5 (Windows, macOS, Linux, Android, iOS, консоли) – настройки проекта (Project Settings) для целевых платформ: разрешение, API рендеринга, управление памятью – профили сборки (Development, Shipping, Debug) и их влияние на производительность и отладку – принципы упаковки и развертывания контента (Asset Manager, Chunking, платформенные сервисы) – архитектуру игровых объектов UE5: Actor, Component, Pawn, Character, Controller, GameMode, GameState	Владеет системой ввода Enhanced Input: Input Actions, Input Mapping Contexts, обработку модов владеет основами Blueprints Visual Scripting и C++ для UE5 (рефлексия, UCLASS, UPROPERTY, UFUNCTION) владеет методами работы с физикой, графикой и системами частиц владеет методами оптимизации графики: LOD, Culling, Instancing, GPU Visualizer владеет методами анализа и тестирования баланса: логирование, симуляции, автоматизация	Устный опрос. Защита результатов работы над проектом.
Перечень умений, осваиваемых в рамках дисциплины		
Уметь: – Настраивать проект под сборку для выбранной платформы (Windows, Android, iOS) – выполнять упаковку проекта (Package Project) с разделением контента на чанки и сжатием – развертывать собранное приложение на тестовых устройствах и в магазинах приложений – создавать игровые объекты (Actor, Character) с компонентами (коллизия, физика, звук) – реализовывать пользовательский ввод через Enhanced Input (привязка клавиш, осей, модов)	Эффективно использует среду разработки; Создает оптимальные вычислительные алгоритмы Грамотно использует анализируемые показатели производительности для улучшения работы проекта Понимает организацию хранения данных	Письменный и устный опрос. Защита результатов работы над проектом.

– программировать игровую логику на Blueprints и C++ (взаимодействие, триггеры, события)		

МИНОБРНАУКИ РОССИИ
ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Колледж информационных и креативных технологий

КОНТРОЛЬНО-ОЦЕНОЧНЫЕ СРЕДСТВА
для проведения текущего контроля и промежуточной аттестации по учебной дисциплине

МДК.06.02. Продвинутая разработка игровых систем Unreal

программы подготовки специалистов среднего звена
09.02.10 Разработка компьютерных игр, дополненной и виртуальной реальности

Форма обучения: очная

Владивосток 2026

Контрольно-оценочные средства для проведения текущего контроля и промежуточной аттестации по учебной дисциплине МДК.06.02. Продвинутая разработка игровых систем Unreal разработаны в соответствии с требованиями ФГОС СПО по специальности / профессии 09.02.10 Разработка компьютерных игр, дополненной и виртуальной реальности, утвержденного приказом Минобрнауки РФ от от 25 июня 2024, № 441, примерной образовательной программой, рабочей программой учебной дисциплины.

Разработчик(и): *О.Ю. Рудзейт, преподаватель*

Рассмотрено и одобрено на заседании цикловой методической комиссии

Протокол № _9_ от «_20_» ___мая____2026_ г.

Председатель ПЦК _____ Н.В. Кравцова
подпись

1. Общие сведения

Контрольно-оценочные средства (далее – КОС) предназначены для контроля и оценки образовательных достижений обучающихся, освоивших программу учебной дисциплины МДК.06.02 «Продвинутая разработка игровых систем Unreal».

КОС включают в себя контрольные материалы для проведения текущего контроля успеваемости и промежуточной аттестации по дисциплине, которая проводится в форме дифференцированного зачёта (с использованием оценочного средства - устный опрос в форме ответов на вопросы билетов, устный опрос в форме собеседования, выполнение письменных заданий, тестирование и т.д.).

2. Планируемые результаты обучения, обеспечивающие результаты освоения образовательной программы

Код ПК	Код результата обучения	Наименование результата обучения
ПК 6.1	З1	Настраивать проект под сборку для выбранной платформы (смена целевой платформы, конфигурация графических настроек).
	З2	Проводить сборку (Package Project) для Windows, Android и iOS с использованием Device Manager.
	З3	анализировать технические задания и выявлять требования к алгоритмам.
	З4	оптимизировать размер и структуру пакетов контента (разделение на чанки, сжатие текстур, выгрузка ненужных ассетов).
	У1	использовать популярные платформы для сборки, настройки и развертывания контента;
	У2	использовать основные платформы сборки проектов на Unreal Engine 5 (Windows, macOS, Linux, Android, iOS, консоли);
	У3	применять настройки проекта (Project Settings) для целевых платформ: разрешение, соотношение сторон, управление памятью, API рендеринга (DirectX 12, Vulkan, Metal);
	У4	работать с профилями сборки (Development, Shipping, Debug) и их влияние на производительность и отладку;
	У5	применять инструменты упаковки контента (Content Browser, Asset Manager, Chunking) для оптимизации дистрибутива;
	У6	работать с системами контроля версий для командной разработки и слияния контента
ПК 6.3	З1	Архитектуру игрового фреймворка Unreal Engine 5: GameMode, GameState, PlayerController, Pawn, Character;
	З2	принципы компонентного подхода и композиции объектов в UE5;
	З3	Реализовывать программные модули на основе требований технического задания.
	З4	систему ввода Enhanced Input System и её настройку под разные устройства;

Код ПК	Код результата обучения	Наименование результата обучения
	З5	механизмы взаимодействия между актерами: интерфейсы, делегаты, события, ноды Communication;
	З6	основы проектирования игровых циклов, состояний и конечных автоматов (State Machines);
	З7	методы балансировки игровой механики на этапе прототипирования;
	З8	паттерны проектирования, применимые в геймдеве (Observer, Command, Strategy, Singleton).
	У1	Разрабатывать механику игрового процесса;
	У2	проектировать и реализовывать базовые и сложные игровые механики с использованием Blueprints и C++;
	У3	настраивать систему ввода для поддержки клавиатуры, мыши, геймпадов и тач-интерфейсов;
	У4	создавать интерактивные системы: инвентарь, квесты, диалоги, боевые механики, физику взаимодействия;
	У5	интегрировать анимации и звуковые эффекты в игровые механики через Animation Notify и Audio Components;
	У6	проводить итеративное тестирование и балансировку механик на основе фидбека и метрик;
	У7	оптимизировать логику геймплея для снижения нагрузки на CPU (минимизация Tick, использование таймеров и событий);
	У8	документировать архитектуру механик для передачи команде и поддержки кода.
ПК 6.4	З1	Архитектуру рендеринга Unreal Engine 5: Deferred Shading, Nanite, Lumen, Virtual Shadow Maps;
	З2	систему частиц и визуальных эффектов: Niagara, Cascade (legacy), GPU Particles;
	З3	методы оптимизации графики: LOD, HLOD, culling, batching, texture streaming;
	З4	работу с пост-процессингом: Volume, Post Process Material, Color Grading, Bloom, DOF;
	З5	принципы работы с освещением: Static/Dynamic/Stationary lights, Lightmass, Light Functions;
	З6	Интеграцию визуальных скриптов и C++ кода для управления графическими системами.
	У1	Программировать игровую графику и специальные эффекты;
	У2	создавать и настраивать сложные материалы и шейдеры для реалистичной и стилизованной графики;
	У3	разрабатывать визуальные эффекты (VFX) в Niagara: взрывы, магия, погода, интерфейсные эффекты;
	У4	оптимизировать графический контент: настраивать LOD, атласы текстур, сжатие, инстансинг;
	У5	настраивать динамическое освещение и тени с учётом производительности целевой платформы;

Код ПК	Код результата обучения	Наименование результата обучения
	У6	проводить профилирование графики (Stat GPU, Unreal Insights) и устранять узкие места рендеринга
ПК 6.5	З1	Теорию игрового баланса: кривые прогрессии, экономику, вероятность, риск/вознаграждение;
	З2	методы математического моделирования баланса: формулы урона, опыта, стоимости, времени;
	З3	инструменты анализа данных в UE5: Stats, Analytics, Replay System, Custom HUD-метрики;
	З4	принципы проектирования внутриигровой экономики: валюты, ресурсы, магазины, награды;
	З5	Методы A/B-тестирования и сбора обратной связи от игроков на ранних этапах;
	З6	Основы поведенческой психологии игрока и её влияние на баланс;
	З7	Инструменты визуализации данных: таблицы (Data Assets), графики.
	У1	Разрабатывать системы игрового баланса;
	У2	проектировать и настраивать формулы баланса с использованием Data-Driven подхода (Data Assets, Tables);
	У3	реализовывать системы отслеживания игровых метрик и аналитики внутри проекта;
	У4	проводить симуляции и итеративную настройку параметров баланса на основе тестовых данных;
	У5	создавать инструменты для дизайнеров: редакторы баланса в редакторе, hot-reload конфигураций;
	У6	балансировать сложность, прогресс и награды с учётом целевой аудитории и платформы;
	У7	интегрировать системы динамической сложности (Dynamic Difficulty Adjustment) при необходимости.

3 Соответствие оценочных средств контролируемым результатам обучения

3.1 Средства, применяемые для оценки уровня теоретической подготовки

Краткое наименование раздела (модуля) / темы дисциплины	Код результата обучения	Показатель[2] овладения результатами обучения	Наименование оценочного средства и представление его в КОС[3]	
			Текущий контроль[4]	Промежуточная аттестация[4]
Раздел 1. Основы архитектуры Unreal Engine 5				
Тема 1.1	З1	Способность понимать и дополнять структуру и состав проекта	Устный опрос (п.5.1 вопр. 1-5)	Собеседование

Краткое наименование раздела (модуля) / темы дисциплины	Код результата обучения	Показатель[2] овладения результатами обучения	Наименование оценочного средства и представление его в КОС[3]	
			Текущий контроль[4]	Промежуточная аттестация[4]
СРС по Теме Введение в архитектуру UE5. Система модулей	З2	Знание и понимание основных понятий: Accet, Actor, Pawn, Interface, Blueprint	Устный опрос (п.5.1 вопр. 6-10)	Собеседование
	У1	Умение ориентироваться в интерфейсе двжика Unreal Engine 5	Устный опрос (п.5.1 вопр. 11-15)	Собеседование
Тема 1.2 СРС по Теме Система типов данных и рефлексия	З3	Способность применять основные типы данных игрового движка	Устный опрос (п.5.1 вопр. 16-20)	Собеседование
	У2	Умение использовать инструменты, готовые ассеты и плагины движка	Устный опрос (п.5.1 вопр. 21-25)	Собеседование
	У3	Способность применять оптимальные типы данных в определенных условиях работы проекта	Устный опрос (п.5.1 вопр. 25-30)	Собеседование
Раздел 2. Система рендеринга и графическая архитектура				
Тема 2.2 СРС по Теме Nanite и виртуальная геометрия	З4	Способность использовать систему Nanite при разработки проекта	Устный опрос (п.5.1 вопр. 31-35)	Собеседование
	У4	Способность определять оптимальное количество полигонов объектов для балансировки графики и производительности	Устный опрос (п.5.1 вопр. 36-40)	Собеседование
Тема 2.3. СРС по Теме Lumen и глобальное освещение	З5	Способность знать основные источники света	Устный опрос (п.5.1 вопр. 41-45)	Собеседование
	З6	Способность использовать встроенные функции движка для улучшения освещения	Устный опрос (п.5.1 вопр. 46-50)	Собеседование
	У5	Умение подобрать оптимальные источники света для увеличения производительности проекта	Устный опрос (п.5.1 вопр. 51-55)	Собеседование
Раздел 3. Системы оптимизации производительности				
Тема 3.1 СРС по Теме Профилирование и анализ производительности	З7	Способность выявлять показатели работы Blueprint’ов в проекте	Устный опрос (п.5.1 вопр. 56-60)	Собеседование
	У6	Способность применять на практике инструменты тестирования производительности	Устный опрос (п.5.1 вопр. 61-65)	Собеседование
Раздел 5. Платформенная специфика и развертывание проекта				
Тема 5.2 СРС по теме	З1	Способность определить основные настройки для оптимизации проекта под сборку на целевую платформу	Устный опрос (п.5.1 вопр. 66-70)	Собеседование

Краткое наименование раздела (модуля) / темы дисциплины	Код результата обучения	Показатель[2] овладения результатами обучения	Наименование оценочного средства и представление его в КОС[3]	
			Текущий контроль[4]	Промежуточная аттестация[4]
Упаковка и развертывание приложений	З2	Способность к выявлению специфики работы проектов на целевых платформах	Устный опрос (п.5.1 вопр. 71-75)	Собеседование
	У1	Способность определить наиболее оптимальные параметры для целевых платформ	Устный опрос (п.5.1 вопр. 76-78)	Собеседование

3.2 Средства, применяемые для оценки уровня практической подготовки

Краткое наименование раздела (модуля) / темы дисциплины	Код результата обучения	Показатель[2] овладения результатами обучения	Наименование оценочного средства и представление его в КОС[3]	
			Текущий контроль[4]	Промежуточная аттестация[4]
Раздел 1. Основы архитектуры Unreal Engine 5				
Тема 1.2 Система типов данных и рефлексия	З1	Способность применять основные типы данных игрового движка	Устный опрос (п.5.1 вопр. 79-81)	Индивидуальный проект
	З2	Умение использовать инструменты, готовые ассеты и плагины движка	Устный опрос (п.5.1 вопр. 82-85)	Индивидуальный проект
	У1	Способность применять оптимальные типы данных в определенных условиях работы проекта	Устный опрос (п.5.1 вопр. 86-88)	Индивидуальный проект
Тема 1.5 Введение в архитектуру UE5. Система модулей	З3	Способность понимать и дополнять структуру и состав проекта	Устный опрос (п.5.1 вопр. 89-92)	Индивидуальный проект
	У2	Знание и понимание основных понятий: Accet, Actor, Pawn, Interface, Blueprint	Устный опрос (п.5.1 вопр. 93-95)	Индивидуальный проект
	У3	Умение ориентироваться в интерфейсе двжика Unreal Engine 5	Устный опрос (п.5.1 вопр. 96-98)	Индивидуальный проект
Раздел 2. Система рендеринга и графическая архитектура				
Тема 2.2 Nanite и виртуальная геометрия	З4	Способность подбирать оптимальные способы визуализации объектов при использовании системы Nanite и без нее	Практическая презентация	Индивидуальный проект
	У4	Способность создавать коллизии с минимальной нагрузкой на ресурсы системы	Практическая презентация	Индивидуальный проект
Тема 2.3. Lumen и	З5	Способность различать источники освещения и определять из них наиболее эффективные	Практическая презентация	Индивидуальный проект

Краткое наименование раздела (модуля) / темы дисциплины	Код результата обучения	Показатель[2] овладения результатами обучения	Наименование оценочного средства и представление его в КОС[3]	
			Текущий контроль[4]	Промежуточная аттестация[4]
глобальное освещение	З6	Способность использовать встроенные функции движка для улучшения освещения	Практическая презентация	Индивидуальный проект
	У5	Способность создавать реалистичное освещение окружающего мира, объектов, персонажа	Практическая презентация	Индивидуальный проект
Раздел 3. Системы оптимизации производительности				
Тема 3.1 Профилирование и анализ производительности	З7	Способность выявлять показатели работы Blueprint’ов в проекте	Практическая презентация	Индивидуальный проект
	У6	Способность применять на практике инструменты тестирования производительности	Практическая презентация	Индивидуальный проект
Тема 3.2 Оптимизация CPU и игровых систем	З8	Способность анализировать способы оптимального распределения нагрузки	Практическая презентация	Индивидуальный проект
	У7	Способность применять в проекте асинхронные методы вычислений в проекте	Практическая презентация	Индивидуальный проект
Раздел 5. Платформенная специфика и развертывание проекта				
Тема 5.2 Упаковка и развертывание приложений	З9	Способность вычислить основные настройки для оптимизации проекта под сборку на целевую платформу	Практическая презентация	Индивидуальный проект
	З10	Способность к выявлению специфики работы проектов на целевых платформах	Практическая презентация	Индивидуальный проект
	У8	Способность определить наиболее оптимальные параметры для целевых платформ	Практическая презентация	Индивидуальный проект
Тема 5.3 Оптимизация под мобильные платформы	З11	Способность определять оптимальные библиотеки и настройке по работе с мобильными платформами	Практическая презентация	Индивидуальный проект
	З12	Способность использовать инструменты для компиляции проекта под целевую мобильную платформу	Практическая презентация	Индивидуальный проект
	У9	Способность исправлять ошибки на этапе компиляции проекта	Практическая презентация	Индивидуальный проект
	У10	Способность выявлять и тестировать функции, которые потенциально могут привести к некорректной работе приложения	Практическая презентация	Индивидуальный проект

4. Описание процедуры оценивания

Результаты обучения по модулю, уровень сформированности компетенций оцениваются по четырёх бальной шкале оценками: «отлично», «хорошо», «удовлетворительно», «неудовлетворительно» (по бальной системе. Максимальная сумма баллов по дисциплине равна 100 баллов.)

Текущая аттестация по модулю проводится с целью систематической проверки достижений обучающихся. Объектами оценивания являются: степень усвоения теоретических знаний, уровень овладения практическими умениями и навыками по всем видам учебной работы, качество выполнения самостоятельной работы, учебная дисциплина (активность на занятиях, своевременность выполнения различных видов заданий, посещаемость всех видов занятий по аттестуемой дисциплине).

При проведении промежуточной аттестации оценивается достижение студентом запланированных по модулю результатов обучения, обеспечивающих результаты освоения образовательной программы в целом. Оценка на зачете / экзамене выставляется с учетом оценок, полученных при прохождении текущей аттестации.

Критерии оценивания устного ответа

(оценочные средства: контрольная работа, собеседование, устное сообщение.)

5 баллов - ответ показывает прочные знания основных процессов изучаемой предметной области, отличается глубиной и полнотой раскрытия темы; владение терминологическим аппаратом; умение объяснять сущность, явлений, процессов, событий, делать выводы и обобщения, давать аргументированные ответы, приводить примеры; свободное владение монологической речью, логичность и последовательность ответа; умение приводить примеры современных проблем изучаемой области.

4 балла - ответ, обнаруживающий прочные знания основных процессов изучаемой предметной области, отличается глубиной и полнотой раскрытия темы; владение терминологическим аппаратом; умение объяснять сущность, явлений, процессов, событий, делать выводы и обобщения, давать аргументированные ответы, приводить примеры; свободное владение монологической речью, логичность и последовательность ответа. Однако допускается одна - две неточности в ответе.

3 балла – ответ, свидетельствующий в основном о знании процессов изучаемой предметной области, отличающийся недостаточной глубиной и полнотой раскрытия темы; знанием основных вопросов теории; слабо сформированными навыками анализа явлений, процессов, недостаточным умением давать аргументированные ответы и приводить примеры; недостаточно свободным владением монологической речью, логичностью и последовательностью ответа. Допускается несколько ошибок в содержании ответа; неумение привести пример развития ситуации, провести связь с другими аспектами изучаемой области.

2 балла – ответ, обнаруживающий незнание процессов изучаемой предметной области, отличающийся неглубоким раскрытием темы; незнанием основных вопросов теории, несформированными навыками анализа явлений, процессов; неумением давать аргументированные ответы, слабым владением монологической речью, отсутствием логичности и последовательности. Допускаются серьезные ошибки в содержании ответа; незнание современной проблематики изучаемой области.

Критерии оценивания письменной работы

(оценочные средства: письменный отчет по практической работе, доклад (сообщение), в том числе выполненный в форме презентации, творческое задание).

5 баллов - студент выразил своё мнение по сформулированной проблеме, аргументировал его, точно определив ее содержание и составляющие. Проблема раскрыта полностью, выводы обоснованы. Приведены данные отечественной и зарубежной литературы, статистические сведения, информация нормативно-правового характера. Студент владеет навыком самостоятельной работы по заданной теме; методами и приемами анализа теоретических и/или практических аспектов изучаемой области. Фактических ошибок, связанных с пониманием проблемы, нет; графически работа оформлена правильно.

4 балла - работа характеризуется смысловой цельностью, связностью и последовательностью изложения; допущено не более 1 ошибки при объяснении смысла или содержания проблемы. Проблема раскрыта. Не все выводы сделаны и/или обоснованы. Для аргументации приводятся данные отечественных и зарубежных авторов. Продемонстрированы исследовательские умения и навыки. Фактических ошибок, связанных с пониманием проблемы, нет. Допущены одна-две ошибки в оформлении работы.

3 балла – студент проводит достаточно самостоятельный анализ основных этапов и смысловых составляющих проблемы; понимает базовые основы и теоретическое обоснование выбранной темы. Проблема раскрыта не полностью. Выводы не сделаны и/или выводы не обоснованы Проведен анализ проблемы без привлечения дополнительной литературы. Допущено не более 2 ошибок в смысле или содержании проблемы, оформлении работы.

2 балла - работа представляет собой пересказанный или полностью переписанный исходный текст без каких бы то ни было комментариев, анализа. Не раскрыта структура и теоретическая составляющая темы. Проблема не раскрыта. Выводы отсутствуют. Допущено три или более трех ошибок в смысловом содержании раскрываемой проблемы, в оформлении работы.

Критерии выставления оценки студенту на зачете/ экзамене

(оценочные средства: устный опрос в форме ответов на вопросы билетов, устный опрос в форме собеседования, выполнение письменных разноуровневых задач и заданий, комплексная расчетно-графическая работа, творческое задание, кейс-задача, портфолио, проект и т.п.)

Оценка по промежуточной аттестации	Характеристика качества сформированности компетенций
«зачтено» / «отлично»	Студент демонстрирует сформированность дисциплинарных компетенций на продвинутом уровне: обнаруживает всестороннее, систематическое и глубокое знание учебного материала, умеет тесно увязывать теорию с практикой, свободно справляется с задачами, вопросами и другими видами применения знаний, причем не затрудняется с ответом при видоизменении заданий, правильно обосновывает принятое решение, владеет разносторонними навыками и приемами выполнения практических задач.

«зачтено» / «хорошо»	Студент демонстрирует сформированность дисциплинарных компетенций на базовом уровне: основные знания, умения освоены, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.
«зачтено» / «удовлетворительно»	Студент демонстрирует сформированность дисциплинарных компетенций на пороговом уровне: имеет знания только основного материала, но не усвоил его деталей, в ходе контрольных мероприятий допускаются значительные ошибки, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала, испытывает затруднения при выполнении практических работ, при оперировании знаниями и умениями при их переносе на новые ситуации.
«не зачтено» / «неудовлетворительно»	Студент демонстрирует сформированность дисциплинарных компетенций на уровне ниже порогового: выявляется полное или практически полное отсутствие знаний значительной части программного материала, студент допускает существенные ошибки, неуверенно, с большими затруднениями выполняет практические работы, умения и навыки не сформированы.

5. Примеры оценочных средств для проведения промежуточной аттестации

5.1 Вопросы для собеседования (устного опроса)

1. Из каких основных папок и файлов состоит проект Unreal Engine 5?
2. Что такое «модуль» в UE5? Как организована система модулей?
3. Как добавить новый модуль в проект? Какие файлы необходимо создать или изменить?
4. Где хранятся настройки проекта (Project Settings) и как они влияют на сборку?
5. Как устроен файл .uproject? Какие параметры можно в нём изменить?
6. Дайте определение понятию «Ассет» в UE5. Приведите примеры типов ассетов.
7. Что такое Actor? Чем Actor отличается от Object?
8. Для чего служит класс Pawn? Какова его роль в игровой архитектуре?
9. Что такое Interface (интерфейс) в контексте Blueprints и C++? Приведите пример использования.
10. Опишите, что такое Blueprint. Какие типы Blueprint’ов существуют (Level Blueprint, GameMode Blueprint и др.)?
11. Перечислите основные окна редактора UE5 (Viewport, Content Browser, Outliner, Details). Назначение каждого.
12. Как создать новый Actor и добавить ему компонент (например, статическую сетку) через интерфейс?
13. Где находится панель управления режимами (Mode Panel) и для чего она используется?
14. Как быстро найти нужный ассет в Content Browser? Какие фильтры и поиск доступны?
15. Как открыть Blueprint Editor? Покажите (опишите) основные элементы его интерфейса.
16. Назовите базовые типы данных в UE5 (int, float, bool, FString, FName, FText). В чём отличие FName от FString?
17. Что такое TArray, TMap, TSet? Приведите примеры использования.
18. Как объявить свойство (UPROPERTY) в C++ для UE5? Какие спецификаторы (EditAnywhere, BlueprintReadWrite и др.) вы знаете?
19. Что такое рефлексия в контексте UE5? Какие макросы обеспечивают рефлексию?
20. Как работает сборщик мусора (Garbage Collector) в UE5? Какие объекты им управляются?
21. Как импортировать готовый 3D-модель (FBX) в проект? Какие настройки импорта важны?

22. Где находится менеджер плагинов (Plugins) в UE5? Как включить плагин, например, Niagara?

23. Как добавить ассет из Marketplace в проект? Какие шаги необходимо выполнить?

24. Что такое Blueprint Function Library? Как её создать и использовать?

25. Как использовать готовые материалы из стартового контента (Starter Content)?

26. Как создать пользовательский тип данных (структуру) в Blueprints?

27. Как выполнить приведение типов (Cast) в Blueprint и в C++?

28. Как объявить событие (Event) и связать его с функцией в Blueprint?

29. Что такое Nanite? Для решения каких задач он был создан?

30. Какие типы геометрии не поддерживаются Nanite?

31. Как включить Nanite для статической меши? Какие ограничения накладывает Nanite на материалы?

32. Как проверить, активен ли Nanite для выбранного объекта в сцене?

33. В каких случаях использование Nanite не даёт выигрыша в производительности?

34. Какой фактор (полигональность, количество объектов, LOD) важнее для производительности при использовании Nanite?

35. С помощью каких инструментов (например, консольных команд) можно оценить нагрузку на GPU от Nanite?

36. Как настроить Fallback Relative Error для Nanite-мешей?

37. Когда стоит отключить Nanite для детализированного объекта?

38. Как сбалансировать качество и производительность при работе с Nanite и традиционными LOD?

39. Перечислите основные типы источников света в UE5 (Directional, Point, Spot, Sky Light, Rect Light).

40. Чем отличается статическое (Static) освещение от динамического (Movable)?

41. Что такое Sky Light и как он использует Cubemap для освещения сцены?

42. Для чего нужен Lightmass Importance Volume?

43. Как работают теневые карты (Shadow Maps) для разных типов света?

44. Что такое Lumen? Как он реализует глобальное освещение и отражения?

45. Как включить Lumen в проекте? Какие настройки пост-обработки влияют на Lumen?

46. Какие преимущества имеет Lumen по сравнению с традиционным Lightmass Baking?

47. Как настроить качество Lumen через консольные команды (r.Lumen.…)

48. Какие артефакты могут возникать при использовании Lumen и как их уменьшить?

49. Какой тип источника света наиболее затратен для производительности? Почему?

50. Какие настройки теней (Shadow Resolution, Distance Field Shadows) критически влияют на FPS?

51. При каких условиях стоит заменить динамический свет на статический (baked)?

52. Как влияет количество источников света на производительность Lumen?

53. Как с помощью Light Channels уменьшить количество пересчётов освещения для разных объектов?

54. Какие инструменты в UE5 позволяют профилировать Blueprint? (Blueprint Profiler, Unreal Insights)

55. Какие метрики важны при анализе производительности Blueprint (время выполнения, количество вызовов, память)?

56. Какую информацию даёт узел «Profile» в Blueprint?

57. Как обнаружить «тяжёлый» Blueprint, который вызывает просадки FPS?

58. Что такое «танцующие кадры» (hitches) и как их связать с конкретными Blueprint-функциями?

59. Как запустить встроенный профилировщик (Session Frontend, Trace)?

60. Как использовать консольную команду stat unit? Что означают показатели «Frame», «Game», «Draw», «GPU»?

61. Как записать трассировку (trace) через Unreal Insights и проанализировать её?

62. Как провести автоматическое нагрузочное тестирование (Automation Testing) для оценки производительности?

63. С помощью какого инструмента можно проверить потребление памяти ассетами (Memory Profiler)?

64. Какие настройки в Project Settings критически важны для сборки под Android (например, ориентация экрана, разрешение, API рендеринга)?

65. Как настроить проект для сборки под Windows с максимальной производительностью (Shipping конфигурация)?

66. Что такое «Primary Asset Labels» и как они помогают оптимизировать загрузку на платформах с ограниченной памятью?

67. Какие параметры сжатия текстур и звуков лучше выбрать для iOS? Почему?

68. Как указать разные карты уровней (Levels) для загрузки на разных платформах?

69. Какие особенности ввода (сенсорный экран, контроллер, мышь) необходимо учитывать при сборке для консолей или мобильных устройств?

70. Как влияет ограничение оперативной памяти на Android на использование Lumen и Nanite?

71. Почему для некоторых платформ (например, Sony Playstation) рекомендуется уменьшать количество динамических источников света?

72. Какие особенности версионирования и патчинга существуют для Steam и Epic Games Store?

73. В чём отличие между Development, DebugGame и Shipping сборками с точки зрения работы на целевой платформе?

74. Как выбрать разрешение текстур и уровень сжатия для мобильной игры на Android с учётом производительности?

75. Как настроить минимальное LOD (Minimum LOD) для объектов на слабых платформах?

76. Какой профиль сборки (Development, Test, Shipping) стоит выбрать для финального развёртывания в магазин приложений?

77. Как уменьшить размер установочного файла (.apk, .ipa) без потери видимого качества?

78. Какие консольные команды или параметры запуска можно задать для оптимизации производительности на целевой платформе?

79. Какие базовые типы данных поддерживаются в UE5 (int, float, bool, FString, FName, FText)? В чём отличие FName от FString?

80. Что такое TArray, TMap, TSet? Приведите примеры, когда использовать каждый из контейнеров.

81. Как объявить свойство (UPROPERTY) в C++ для UE5? Какие спецификаторы (EditAnywhere, BlueprintReadWrite, VisibleDefaultsOnly и др.) вы знаете и для чего они нужны?

82. Объясните понятие «рефлексия» в контексте Unreal Engine. Каким образом и с помощью каких макросов она реализуется?

83. Как работает сборщик мусора (Garbage Collector) в UE5? Какие объекты подлежат автоматическому удалению, а какие требуют ручного управления?

84. Что такое UPROPERTY и UFUNCTION? Какие мета-спецификаторы можно использовать для функций?

85. Как создать пользовательскую структуру (struct) с рефлексией в UE5? В чём отличие структуры от класса (UCLASS) с точки зрения рефлексии?

86. Что такое UPackage и как связаны пакеты с сериализацией объектов?

87. Как выполнить приведение типов (Cast) в Blueprint и в C++? В чём разница между Cast<> и CastChecked<>?

88. Как объявить Blueprint-функцию в C++ и сделать её доступной из графа событий?

89. Перечислите основные архитектурные элементы проекта UE5: Asset, Actor, Pawn, Character, Controller, GameMode, GameState. Какова их роль?

90. Что такое «модуль» в UE5? Из каких файлов состоит модуль (.Build.cs, .h, .cpp)?

91. Как добавить новый модуль в существующий проект? Какие изменения нужно внести в файл .uproject?

92. Опишите структуру каталогов стандартного проекта UE5 (Config, Content, Source, Intermediate,

Saved). Какое назначение каждой папки?

93. Что такое GameInstance и чем она отличается от GameMode? Когда используется GameInstance?
94. Как организован цикл работы класса Engine Loop (инициализация, обновление, завершение)?
95. В чём разница между Level Blueprint, GameMode Blueprint и Actor Blueprint?
96. Как происходит загрузка уровней (Level Streaming) и какие объекты при этом выгружаются?
97. Что такое World и Level в UE5? Как получить текущий мир из C++?
98. Как работают Subsystems (GameInstanceSubsystem, WorldSubsystem, LocalPlayerSubsystem)? Приведите пример использования