

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение высшего образования
«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Факультет компьютерных технологий и прикладной математики
Центр искусственного интеллекта

УТВЕРЖДАЮ:

Проректор по учебной работе, качеству образования – первый проректор

_____ Хагуров Т.А.

подпись

«02» июня 2026



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
Б1.О.26 КОМПЬЮТЕРНАЯ ГРАФИКА

Направление подготовки 02.03.03 Математическое обеспечение и администрирование информационных систем

Направленность (профиль) Искусственный интеллект и аналитика данных

Форма обучения очная

Квалификация бакалавр

Краснодар 2026

Рабочая программа дисциплины «Введение в специальность» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования (ФГОС ВО) по направлению подготовки 02.03.03 Математическое обеспечение и администрирование информационных систем.

Программу составили:

А.В. Коваленко, заведующий КАДИИ,
д-р.тех.н., профессор


подпись

А.С. Мисулев
Ведущий рендер-программист
ООО «Кар Икс Технолоджис»


подпись

Е.Н. Калайдин, доктор физ.-мат.наук,
профессор


подпись

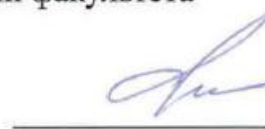
Г.В. Калайдина, канд. физ.-мат.наук,
доцент


подпись

Рабочая программа дисциплины утверждена на заседании центра
искусственного интеллекта
протокол № 4 «14» мая 2026 г.
Руководитель центра ИИ Коваленко А.В.


подпись

Утверждена на заседании учебно-методической комиссии факультета
компьютерных технологий и прикладной математики
протокол № 3 «15» мая 2026 г.
Председатель УМК факультета Добровольская Н.Ю.


подпись

Экспертиза:

Экспертная группа индустриального партнера ООО «CarX Technologies»:
Фокин Д.Н., Генеральный директор, ceo@carx-tech.com; Щеголев А.В.,
Ведущий инженер-разработчик, a.shchegolev@carx-tech.com; Болоцкий Д.Д.,
DevOps-инженер, d.bolotskiy@carx-tech.com.

Рецензенты:

Мостовой Евгений Викторович, генеральный директор ООО «Портал-Юг»,
e-mail: mostovoy@portal-yug.ru

Луценко Евгений Вениаминович, доктор экономических наук, кандидат
технических наук, профессор кафедры компьютерных технологий и систем
ФГБОУ ВО «Кубанский государственный аграрный университет имени И.Т.
Трубилина», e-mail: prof.lutsenko@gmail.com

1 Цели и задачи изучения дисциплины (модуля)

1.1 Цель освоения дисциплины

Формирование у обучающихся теоретических знаний и практических навыков в области компьютерной графики, включая изучение аппаратных и программных средств, основных алгоритмов и технологий, а также приобретение опыта разработки интерактивных графических приложений с использованием среды Unity и обоснованного выбора методов визуализации данных на основе их топологических и метрических свойств

(МИП – материалы индустриального партнера интегрированные в РПД)

Дополнительно с участием индустриального партнёра (МИП):

Освоение современных подходов к разработке высокопроизводительных графических решений на основе шейдеров (PBR-освещение, карты нормалей, оптимизация для мобильных GPU) в контексте реальных задач игровой индустрии.

1.2 Задачи дисциплины

1. Изучить принципы работы графических систем, форматы данных и графические библиотеки.
2. Освоить среду разработки Unity, её инструменты и компоненты.
3. Научиться создавать 2D/3D объекты, настраивать анимацию, поверхности, камеру и игровую область.
4. Получить навыки программирования на C# для управления графическими объектами (перемещение, коллизии, разрушение, счётчики).
5. Сформировать компетенции, необходимые для визуализации данных и разработки приложений с элементами ИИ.
6. Освоить физически корректное освещение (PBR), работу с картами нормалей и оптимизацию шейдеров для мобильных платформ (МИП).

1.3 Место дисциплины (модуля) в структуре образовательной программы

Дисциплина «Компьютерная графика» относится к обязательной части Блока 1 "Дисциплины (модули)" учебного плана. В соответствии с рабочим учебным планом дисциплина изучается на 2 курсе по очной форме обучения. Вид промежуточной аттестации: зачет.

Пререквизиты: (Формируют инфраструктуру для освоения базовых уровней компетенций дисциплины) Алгебра и геометрия, Основы программирования, Алгоритмы и структуры данных, Математический анализ.

Постреквизиты (не содержат развития компетенций на следующие уровни средний, продвинутый): Разработка мобильных приложений, Генеративный ИИ, WEB-разработка.

1.4 Профессиональные роли в структуре образовательной программы

Роль 1: Data Analyst (Аналитик данных)

Извлечение знаний из данных, построение аналитических моделей, использующих МО и ИИ.

Задачи:

1. Статистический анализ, визуализация данных, предварительная обработка.
2. Создание прогнозных моделей
3. Построение аналитических моделей для поддержки бизнес-решений.

Роль 2: MLOps (Специалист по эксплуатации ИИ)

Автоматизация и операционное управление жизненным циклом МО-моделей

Задачи:

1. DevOps для ML.
2. Автоматизация, мониторинг ML-систем.
3. Операционное управление жизненным циклом ML-моделей.

Роль 3: AI PM (Менеджер проектов ИИ)

Управление процессами создания ИИ-решений, включая координацию команды разработки

Задачи:

1. Управление ИИ-проектами от идеи до внедрения
2. Анализ бизнес-требований и постановка задач
3. Оценка эффективности и ROI ИИ-решений

1.5 Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы

Изучение данной учебной дисциплины направлено на формирование у обучающихся следующих компетенций:

Код и наименование индикатора достижения компетенции	Результаты обучения по дисциплине
ОПК-5. Способен устанавливать и сопровождать программное обеспечение для информационных систем и баз данных, в том числе отечественного производства	
ОПК-5.2 Осуществляет установку, настройку и техническое сопровождение программного и аппаратного обеспечения для информационных и автоматизированных систем	Знает: методы установки и конфигурации графических сред (Unity, рендер-пайплайны Built-in, URP, HDRP), способы настройки аппаратного обеспечения (GPU, драйверы) и ПО для визуализации; инструменты диагностики графического конвейера (Frame Debugger, Profiler); основы инсталляции шейдеров, текстурных ассетов и библиотек (в т.ч. отечественных аналогов). (МИП: настройка Universal Render Pipeline для мобильных платформ, интеграция PBR-материалов, оптимизация draw calls.) Умеет: устанавливать и настраивать среду Unity, импортировать и сопровождать графические активы (модели, текстуры, шейдеры); конфигурировать параметры качества, освещения, теней; выполнять диагностику производительности графической подсистемы с использованием встроенных профилировщиков; адаптировать рендер-пайплайн под целевое железо (мобильные GPU, десктоп). (МИП: настройка тестовой среды с URP, диагностика через Frame Debugger, оптимизация шейдеров для мобильных устройств.) Владеет: навыками инсталляции и сопровождения графических проектов Unity (управление зависимостями, обновление версий); приёмами настройки визуальных эффектов (шейдеры, карты нормалей, пост-обработка) под требования производительности; методами технической поддержки графических информационных систем (логирование, профилирование, устранение артефактов). (МИП: сопровождение PBR-шейдеров в реальных проектах игровой индустрии, настройка уровней детализации (LOD) и батчинга.)
PL-3 Способен применять языки программирования платформы .NET для решения задач в области ИИ	
PL-3.1 – Разрабатывает и отлаживает прикладные решения с элементами ИИ на C#	Знает: основы .NET API для работы с графикой, математическими вычислениями и коллекциями; способы реализации простых адаптивных алгоритмов (например, счётчик очков, перемещение объекта к цели). Умеет: создавать скрипты-обработчики на C#, реализующие логику поведения объектов (слежение камеры, спавн объектов, изменение скорости). Владеет: приёмами разработки на C# в среде Visual Studio/Rider с последующей компиляцией в Unity; методами отладки и профилирования кода, содержащего элементы ИИ (например, простое управление через ИИ-агента)
MF-7 Способен применять методы дифференциальной геометрии и топологии для формализации, анализа и интерпретации структур данных и признаков пространств, включая задачи отображения, кластеризации, обучения на многообразиях и анализа устойчивости моделей.	

Код и наименование индикатора достижения компетенции	Результаты обучения по дисциплине
MF 7.3 – Обосновывает выбор методов визуализации или трансформации признаков с учётом их топологических и метрических свойств.	Знает: математические основы работы с координатами (мировые, экранные, GUI), аффинные преобразования, матрицы трансформации, метрики расстояний (евклидова, манхэттенская) в контексте расположения объектов на сцене.
	Умеет: использовать системы координат Unity для позиционирования графических объектов, настраивать камеру для оптимального отображения признаков пространств (например, 2D-проекция многомерных данных).
	Владет: навыками выбора подходящего типа визуализации (точки, поверхности, цветовые карты) в зависимости от топологической структуры данных (кластеры, многообразия) и метрических свойств (разреженность, плотность).

Результаты обучения по дисциплине достигаются в рамках осуществления всех видов контактной и самостоятельной работы обучающихся в соответствии с утвержденным учебным планом.

Индикаторы достижения компетенций считаются сформированными при достижении соответствующих им результатов обучения.

2. Структура и содержание дисциплины

2.1 Распределение трудоёмкости дисциплины по видам работ

Общая трудоёмкость дисциплины составляет 2 зачетных единиц (72 часа), их распределение по видам работ представлено в таблице

Виды работ	Всего часов	Форма обучения очная
		4 семестр (часы)
Контактная работа, в том числе:	50,2	50,2
Аудиторные занятия (всего):	48	48
занятия лекционного типа	16	16
лабораторные занятия	32	32
Иная контактная работа:	2,2	2,2
Контроль самостоятельной работы (КСР)	2	2
Промежуточная аттестация (ИКР)	0,2	0,2
Самостоятельная работа, в том числе:	21.8	21.8
<i>Самостоятельное изучение разделов, самоподготовка (проработка и повторение лекционного материала и материала учебников и учебных пособий, подготовка к лабораторным и практическим занятиям, коллоквиумам и т.д.)</i>	21,8	21,8
Контроль:		
Подготовка к экзамену		
Общая трудоемкость	час.	72
	в том числе контактная работа	50,2
	зач. ед	2

2.2 Содержание дисциплины

Распределение видов учебной работы и их трудоемкости по разделам дисциплины. Разделы (темы) дисциплины, изучаемые в 4 семестре (очная форма обучения)

№	Наименование разделов (тем)	Количество часов			
		Всего	Аудиторная работа		Внеаудиторная работа
			Л	ЛР	
Блок 1. Базовые разделы					
1	Введение в компьютерную графику	3,8	2	–	1,8
2	Среда разработки Unity. Инструменты, настройка проекта	8	2	4	2
3	Компоненты GameObjects. Система событий	8	2	4	2
4	Координаты в GUI Unity 3D	8	–	4	4
5	Анимация (классическая)	8	–	4	4
6	Настройка поверхности (базовая)	6	–	4	2
7	Камера и игровая область	4	2	–	2
8	Скрипт-файл сценария игры (основы C# в Unity).	8	2	4	2
9	Игровое меню. Счётчик очков	4	2	-	2
Блок 2. Материалы индустриального партнёра (МИП)					
10	Архитектура графического конвейера и рендер-пайплайны в Unity	2	2	–	–
11	Математический аппарат трёхмерной графики: векторная алгебра, матричные преобразования и проекции	2	2	–	–
12	Конфигурация тестовой среды. Интеграция материалов, моделей, текстур. Диагностика графического конвейера	2	–	2	–
13	Проектирование и реализация базового пользовательского шейдера	2	–	2	–
14	Алгоритмы вершинной деформации и процедурной анимации в шейдерах	2	–	2	–
15	Генерация процедурных текстурных паттернов на фрагментном этапе	2	–	2	–
	ИТОГО по разделам дисциплины	69,8	16	32	21,8
	Контроль самостоятельной работы (КСР)	2			
	Промежуточная аттестация (ИКР)	0,2			
	Подготовка к текущему контролю				
	Общая трудоемкость по дисциплине	72			

Примечание: Л – лекции, ЛР – лабораторные занятия, СРС – самостоятельная работа студента

2.3 Содержание разделов (тем) дисциплины

2.3.1 Занятия лекционного типа

№	Наименование раздела (темы)	Содержание раздела (темы)	Соответствие индикаторам компетенций	Форма текущего контроля Пояснение
Базовые лекции				
1	Введение в компьютерную графику	Аппаратная база машинной графики (видеокарты, дисплеи). Программные средства: растровые и векторные редакторы. Графические языки и библиотеки (OpenGL, DirectX). Графические возможности языков высокого уровня.	МФ-7.3 (частично)	Устный опрос / тест с выбором ответа (5–7 мин)
2	Среда разработки Unity. Инструменты, настройка проекта	Интерфейс Unity (окна Scene, Game, Hierarchy, Project, Inspector). Создание нового проекта, настройка основных параметров. Импорт стандартных активов. Организация сцены (объекты, компоненты).	ОПК-5.2, PL-3.1	Практическое задание в аудитории (короткий чек-лист)

3	Компоненты GameObjects. Система событий	Понятие GameObject и компонентов (Transform, MeshRenderer, Collider, Rigidbody). Система событий (EventSystem). Создание компонента «падающий куб» с гравитацией и коллайдером.	ОПК-5.2, PL-3.1	Письменный опрос (2–3 вопроса с кратким ответом)
4	Камера и игровая область	Настройка камеры (слежение за объектом, перспектива). Компонент Transform для управления положением камеры. Выбор проекции (перспективная / ортографическая) для отображения признаков пространств.	MF-7.3	Практическая задача: «Для отображения кластера точек в 3D выберите тип проекции и обоснуйте»
5	Скрипт-файлы сценария игры	Разработка на C# скриптов: реакция на столкновение (уничтожение объекта), разрушение при касании, перемещение игрового объекта за указателем мыши. Стандарты оформления кода.	ОПК-5.2, PL-3.1	Контрольная работа – написание небольшого фрагмента кода (например, «OnCollisionEnter уничтожает объект»)
6	Игровое меню. Счётчик очков	Создание игрового меню (кнопки Start/Exit). Создание счётчика очков с прибавлением при подборе предметов. Элементы адаптивного ИИ (сбор статистики, изменение скорости).	PL-3.1	Устное собеседование / защита мини-проекта
Лекции индустриального партнёра (МИП) – 4 часа				
7	Архитектура графического конвейера и рендер-пайплайны в Unity (МИП)	Этапы графического конвейера: вершинный, геометрический, фрагментный этапы, растеризация, тестирование глубины, blending. Роль шейдеров (vertex, fragment). Понятие рендер-пайплайна: Built-in, URP, HDRP. Особенности Universal Render Pipeline (URP): настройка, преимущества для мобильных платформ. Сравнение производительности.	ОПК-5.2 (знание методов настройки и отладки рендер-пайплайна), MF-7.3 (понимание преобразований на этапах конвейера)	Устный опрос (5–7 мин) с вопросами: «Назовите этапы конвейера», «Чем URP отличается от Built-in?»; тест с выбором ответа (в конце лекции)
8	Математический аппарат трёхмерной графики: векторная алгебра, матричные преобразования и проекции (МИП)	Векторы: сложение, скалярное произведение (вычисление угла падения света $N \cdot L$), векторное произведение (вычисление битангента). Матрицы: преобразования (перенос, поворот, масштабирование), матрица модели, вида, проекции. Системы координат: локальная, мировая, видовая, клип-пространство. Перспективная и ортографическая проекции. Композиция матриц (MVP).	MF-7.3 (обоснование выбора систем координат, метрики расстояний)	Решение задач на преобразование координат (письменно, 10 мин): например, «Переведите точку из мировых координат в экранные», «Вычислите скалярное произведение нормали и направления света»

Рекомендуемый регламент применения формы текущего контроля

- Устный опрос / тест – на 5–7 минут в конце лекции (или в начале следующей).
- Практическое задание (короткое) – выполняется в аудитории под контролем преподавателя, 5–10 минут.
- Письменный опрос / решение задачи – 10 минут.

- Контрольная работа (код) – выносится отдельным занятием (например, после лекции 8) или выполняется как домашнее задание с последующей проверкой.
- Мини-кейс / обоснование – можно проводить в форме блиц-опроса или письменного аргументированного ответа.

2.3.2 Лабораторные занятия

№ ЛР	Наименование темы (Тематика лабораторной работы)	Содержание лабораторной работы	Индикатор компетенции	Форма текущего контроля
Базовые лабораторные работы				
1	Среда разработки Unity. Инструменты, настройка проекта	1. Создание нового 3D-проекта в Unity. 2. Настройка основных окон (Scene, Game, Hierarchy, Project, Inspector). 3. Импорт стандартных ассетов (Standard Assets) и простых моделей. 4. Создание первой сцены: размещение объектов (куб, сфера, плоскость). 5. Настройка освещения (Directional Light) и материала по умолчанию. 6. Компиляция и запуск сцены.	ОПК-5.2 (интеграция ассетов, настройка среды), MF-7.3 (выбор системы координат)	Защита ЛР (демонстрация работающей сцены, ответы на вопросы по интерфейсу и настройкам)
2	Компоненты GameObjects. Система событий	1. Добавление компонентов к GameObject (Transform, Mesh Renderer, Collider, Rigidbody). 2. Реализация «падающего куба» с гравитацией и коллайдером. 3. Настройка триггеров и событий (OnTriggerEnter, OnCollisionEnter). 4. Создание простого скрипта на C# для вывода сообщения в консоль при столкновении. 5. Подключение скрипта к объекту и тестирование.	ОПК-5.2 (написание и интеграция кода), PL-3.1 (прикладное решение с обработкой событий)	Защита ЛР (работающий скрипт, демонстрация коллизий, ответы на вопросы о компонентах)
3	Координаты в GUI Unity 3D	1. Изучение мировых, экранных и GUI-координат. 2. Реализация привязки UI-элемента (Text) к позиции 3D-объекта (использование Camera.main.WorldToScreenPoint()). 3. Преобразование координат мыши в мировые (ScreenToWorldPoint). 4. Создание скрипта, отображающего название объекта при наведении курсора. 5. Обоснование выбора системы координат для конкретной задачи.	MF-7.3 (обоснование выбора координат и метрик)	Защита ЛР (демонстрация привязки UI к 3D-объекту, письменное обоснование выбора координат)
4	Анимация (классическая)	1. Создание простой анимации вращения куба с помощью Animation Window (ключевые кадры).	PL-3.1 (разработка анимации на C#, адаптивное поведение)	Защита ЛР (демонстрация анимированного объекта, листинг скрипта, ответы на вопросы)

		2. Настройка параметров анимации (скорость, цикличность). 3. Создание анимации движения по траектории (синусоида, парабола) с помощью скриптов на C# (Update, Time.deltaTime). 4. Запуск анимации по событию (нажатие клавиши). 5. Сравнение анимации через Animation Window и через скрипты.		
5	Настройка поверхности (базовая)	1. Создание материала (Material) с текстурой травы. 2. Настройка параметров материала (Metallic, Smoothness, Tiling). 3. Применение материала к элементу Ground (плоскость). 4. Добавление карты нормалей (Normal Map) для создания рельефа. 5. Настройка отражающей способности (эффект мокрой поверхности). 6. Обоснование выбора текстур и параметров для визуализации высот или плотности данных.	MF-7.3 (выбор визуализации с учётом топологических свойств)	Защита ЛР (сцена с настроенной поверхностью, обоснование выбора материала)
6	Скрипт-файлы сценария игры	1. Разработка скрипта на C# для перемещения объекта за указателем мыши (по плоскости). 2. Реализация обработки столкновений (OnCollisionEnter): уничтожение объекта-цели, увеличение счётчика. 3. Добавление счётчика очков (UI Text) и отображение на экране. 4. Реализация условия победы при достижении определённого количества очков. 5. Отладка и профилирование (Unity Profiler). 6. Интеграция скрипта в проект с соблюдением стандартов кода (комментарии, именование).	ОПК-5.2 (код готов к интеграции), PL-3.1 (прикладное решение с адаптивным счётём)	Защита ЛР (полностью работающий игровой прототип, листинг кода, ответы на вопросы по отладке)
Лабораторные работы индустриального партнёра (МИП) – 2 часа каждая				
7	Конфигурация тестовой среды. Интеграция и настройка материалов, моделей и текстур. Диагностика графического конвейера (МИП)	1. Создание проекта Unity с URP. Настройка линейного цветового пространства. 2. Импорт моделей (FBX) и текстур (диффузные, карты нормалей). 3. Создание материалов с назначением текстур и настройкой параметров (металличность, шероховатость).	ОПК-5.2 (настройка проекта Unity с URP, импорт ассетов, использование Frame Debugger и Profiler для технического	Защита ЛР (демонстрация сцены с несколькими материалами, скриншоты Frame Debugger, ответы на вопросы)

		4. Использование Frame Debugger для анализа вызовов отрисовки (draw calls), этапов шейдера, переключений рендер-состояний. 5. Профилирование через Profiler (GPU Usage, Render Thread).	сопровождения), MF-7.3 (диагностика конвейера)	
8	Проектирование и реализация базового пользовательского шейдера (МИП)	1. Создание шейдерного файла (ShaderLab/HLSL) в Unity. 2. Определение структур appdata (позиция, нормаль, тангент, UV) и v2f (позиция в клип-пространстве, UV, TBN-матрица). 3. Реализация вершинного шейдера: передача позиции через UnityObjectToClipPos, построение TBN-матрицы (tangent, binormal, normal). 4. Реализация фрагментного шейдера: сэмплирование карты нормалей, распаковка UnpackNormal, преобразование вектора света в касательное пространство. 5. Вычисление диффузного освещения (Lambert) с использованием $N \cdot L$.	ОПК-5.2 (код шейдера готов к интеграции), PL-3.1 (прикладное решение с освещением)	Защита ЛР (предоставить .shader файл с комментариями, продемонстрировать работу на модели с нормал-мапой, ответы на вопросы о TBN)
9	Алгоритмы вершинной деформации и процедурной анимации в шейдерах (МИП)	1. Модификация вершинного шейдера для смещения вершин во времени (использование <code>_Time.y</code>). 2. Реализация волнообразной деформации: смещение по $Y = \sin(\text{vertex.x} * \text{частота} + \text{время}) * \text{амплитуда}$. 3. Создание эффекта «подвески» (для колёс): смещение вершин по Y в зависимости от расстояния до центра колеса. 4. Процедурная анимация без использования скелетной анимации – полностью на GPU. 5. Тестирование производительности: сравнение CPU-анимации и вершинного шейдера.	PL-3.1 (реализация адаптивного поведения – анимация на шейдере), MF-7.3 (выбор метода деформации с учётом топологии сетки)	Защита ЛР (демонстрация анимированной сетки, предоставление кода, ответы на вопросы об ограничениях вершинной анимации)
10	Генерация процедурных текстурных паттернов на фрагментном этапе (МИП)	1. Создание фрагментного шейдера без входной текстуры – генерация цвета на основе UV-координат. 2. Реализация полосатого паттерна (чередование цветов по U). 3. Генерация шума Перлина (встроенная функция или ручная реализация). 4. Создание текстуры «асфальт»: комбинация шума + мелкие детали.	MF-7.3 (выбор визуализации – процедурная генерация вместо текстур)	Защита ЛР (демонстрация шейдера на плоскости, скриншоты различных паттернов, ответы на вопросы об эффективности процедурных текстур)

		5. Генерация текстуры «грязь»: наложение пятен с помощью синуса/косинуса от UV. 6. Настройка параметров через свойства шейдера (цвет, масштаб, интенсивность).		
--	--	---	--	--

Защита лабораторной работы (ЛР), выполнение курсового проекта (КП), курсовой работы (КР), расчетно-графического задания (РГЗ), написание реферата (Р), эссе (Э), коллоквиум (К), тестирование (Т) и т.д.

2.3.3 Примерная тематика курсовых работ (проектов)

Не предусмотрены учебным планом

2.4 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)

№	Вид СРС	Перечень учебно-методического обеспечения дисциплины по выполнению самостоятельной работы
1	2	3
1	Изучение теоретического материала	Методические указания по организации самостоятельной работы студентов, утвержденные кафедрой информационных технологий, протокол №1 от 30.08.2019
2	Решение задач	Методические указания по организации самостоятельной работы студентов, утвержденные кафедрой информационных технологий, протокол №1 от 30.08.2019

Учебно-методические материалы для самостоятельной работы обучающихся из числа инвалидов и лиц с ограниченными возможностями здоровья (ОВЗ) предоставляются в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа,
- в форме аудиофайла,
- в печатной форме на языке Брайля.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа,
- в форме аудиофайла.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

3. Образовательные технологии, применяемые при освоении дисциплины (модуля)

В соответствии с требованиями ФГОС в программа дисциплины предусматривает использование в учебном процессе следующих образовательных технологий: чтение лекций с использованием мультимедийных технологий; метод малых групп, разбор практических задач и кейсов.

При обучении используются следующие образовательные технологии:

– Технология коммуникативного обучения – направлена на формирование коммуникативной компетентности студентов, которая является базовой, необходимой для адаптации к современным условиям межкультурной коммуникации.

– Технология разноуровневого (дифференцированного) обучения – предполагает осуществление познавательной деятельности студентов с учётом их индивидуальных способностей, возможностей и интересов, поощряя их реализовывать свой творческий потенциал. Создание и использование диагностических тестов является неотъемлемой частью данной технологии.

– Технология модульного обучения – предусматривает деление содержания дисциплины на достаточно автономные разделы (модули), интегрированные в общий курс.

– Информационно-коммуникационные технологии (ИКТ) - расширяют рамки образовательного процесса, повышая его практическую направленность, способствуют интенсификации самостоятельной работы учащихся и повышению познавательной активности. В рамках ИКТ выделяются 2 вида технологий:

– Технология использования компьютерных программ – позволяет эффективно дополнить процесс обучения языку на всех уровнях.

– Интернет-технологии – предоставляют широкие возможности для поиска информации, разработки научных проектов, ведения научных исследований.

– Технология индивидуализации обучения – помогает реализовывать личностно-ориентированный подход, учитывая индивидуальные особенности и потребности учащихся.

– Проектная технология – ориентирована на моделирование социального взаимодействия учащихся с целью решения задачи, которая определяется в рамках профессиональной подготовки, выделяя ту или иную предметную область.

– Технология обучения в сотрудничестве – реализует идею взаимного обучения, осуществляя как индивидуальную, так и коллективную ответственность за решение учебных задач.

– Игровая технология – позволяет развивать навыки рассмотрения ряда возможных способов решения проблем, активизируя мышление студентов и раскрывая личностный потенциал каждого учащегося.

– Технология развития критического мышления – способствует формированию разносторонней личности, способной критически относиться к информации, умению отбирать информацию для решения поставленной задачи.

Комплексное использование в учебном процессе всех вышеназванных технологий стимулируют личностную, интеллектуальную активность, развивают познавательные процессы, способствуют формированию компетенций, которыми должен обладать будущий специалист.

Основные виды интерактивных образовательных технологий включают в себя:

– работа в малых группах (команде) - совместная деятельность студентов в группе под руководством лидера, направленная на решение общей задачи путём творческого сложения результатов индивидуальной работы членов команды с делением полномочий и ответственности;

– проектная технология - индивидуальная или коллективная деятельность по отбору, распределению и систематизации материала по определенной теме, в результате которой составляется проект;

– анализ конкретных ситуаций - анализ реальных проблемных ситуаций, имевших место в соответствующей области профессиональной деятельности, и поиск вариантов лучших решений;

– развитие критического мышления – образовательная деятельность, направленная на развитие у студентов разумного, рефлексивного мышления, способного выдвинуть новые идеи и увидеть новые возможности.

Подход разбора конкретных задач и ситуаций широко используется как преподавателем, так и студентами во время лекций, лабораторных занятий и анализа результатов самостоятельной работы. Это обусловлено тем, что при исследовании и решении каждой конкретной задачи имеется, как правило, несколько методов, а это требует разбора и оценки целой совокупности конкретных ситуаций.

Темы, задания и вопросы для самостоятельной работы призваны сформировать навыки поиска информации, умения самостоятельно расширять и углублять знания, полученные в ходе лекционных и практических занятий.

Подход разбора конкретных ситуаций широко используется как преподавателем, так и студентами при проведении анализа результатов самостоятельной работы.

Для лиц с ограниченными возможностями здоровья предусмотрена организация консультаций с использованием электронной почты.

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа.

Для лиц с ограниченными возможностями здоровья предусмотрена организация консультаций с использованием электронной почты.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

4. Оценочные средства для текущего контроля успеваемости и промежуточной аттестации

Оценочные средства предназначены для контроля и оценки образовательных достижений обучающихся, освоивших программу учебной дисциплины «Компьютерная графика».

Структура оценочных средств для текущей и промежуточной аттестации

№ п/п	Код и наименование индикатора (в соответствии с п. 1.4)	Результаты обучения (в соответствии с п. 1.4)	Наименование оценочного средства	
			Текущий контроль	Промежуточная аттестация
1.	ОПК-5.2 Осуществляет установку, настройку и техническое сопровождение программного и аппаратного обеспечения для информационных и автоматизированных систем	Знает: методы установки и конфигурации графических сред (Unity, рендер-пайплайны Built-in, URP, HDRP), способы настройки аппаратного обеспечения (GPU, драйверы) и ПО для визуализации; инструменты диагностики графического конвейера (Frame Debugger, Profiler); основы инсталляции шейдеров, текстурных ассетов и библиотек (в т.ч. отечественных аналогов). (МИП: настройка Universal Render Pipeline	Лабораторные работы №№ 1,2,6 (базовые), Лабораторные работы МИП №№ 7,8; Типовые задачи из разделов «Компоненты GameObjects», «Скрипты-файлы»	Кейс №2, Кейс №4 (МИП)

		для мобильных платформ, интеграция PBR-материалов, оптимизация draw calls.) Умеет: устанавливать и настраивать среду Unity, импортировать и сопровождать графические активы (модели, текстуры, шейдеры); конфигурировать параметры качества, освещения, теней; выполнять диагностику производительности графической подсистемы с использованием встроенных профилировщиков; адаптировать рендер-пайплайн под целевое железо (мобильные GPU, десктоп). (МИП: настройка тестовой среды с URP, диагностика через Frame Debugger, оптимизация шейдеров для мобильных устройств.) Владеет: навыками инсталляции и сопровождения графических проектов Unity (управление зависимостями, обновление версий); приёмами настройки визуальных эффектов (шейдеры, карты нормалей, пост-обработка) под требования производительности; методами технической поддержки графических информационных систем (логирование, профилирование, устранение артефактов). (МИП: сопровождение PBR-шейдеров в реальных проектах игровой индустрии, настройка уровней детализации (LOD) и батчинга.)		
2.	PL-3.1 Разрабатывает и отлаживает прикладные решения с элементами ИИ на C#	Знает: основы .NET API для работы с графикой, математическими вычислениями, способы реализации адаптивных алгоритмов. Умеет: создавать скрипты-обработчики, реализующие логику поведения объектов.	- Лабораторные работы №№ 4,6 (базовые) - Лабораторные работы МИП №№ 8,9 - Типовые задачи из разделов «Анимация», «Скрипты-файлы», «Игровое меню»	Кейс №2, Кейс №4 (МИП)

		Владеет: приёмами разработки на C# в Unity, методами отладки и профилирования кода с элементами ИИ.		
3.	MF-7.3 Обосновывает выбор методов визуализации или трансформации признаков с учётом их топологических и метрических свойств	Знает: математические основы работы с координатами, аффинные преобразования, матрицы, метрики расстояний. Умеет: использовать системы координат Unity, настраивать камеру для оптимального отображения признаков пространств. Владеет: навыками выбора типа визуализации в зависимости от топологии данных.	- Лабораторные работы №№ 3,5 (базовые) - Лабораторные работы МИП №№ 7,10 - Типовые задачи из разделов «Координаты», «Настройка поверхности», «Камера»	Кейс №1, Кейс №3, Кейс №4 (МИП)

4.1 Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

Задачи по теме «Компоненты GameObject. Система событий».

Для каждой задачи необходимо реализовать отдельный компонент (скрипт) и прикрепить его к соответствующему игровому объекту. Проверить работоспособность компонентов можно в режиме Play, нажав клавишу «пробел».

1. Добавьте к игровому объекту компонент Rigidbody, чтобы он начал подчиняться физике. Проверьте, что объект падает под действием гравитации.

2. Создайте кнопку UI. Реализуйте событие, которое при нажатии на кнопку меняет цвет выбранного GameObject.

3. Добавьте компонент Collider к игровому объекту и реализуйте событие OnCollisionEnter, чтобы при столкновении с другим объектом появлялось сообщение в консоли.

4. Создайте несколько объектов с компонентом AudioSource. Реализуйте событие, чтобы при клике мышью по объекту воспроизводился уникальный звук.

5. Добавьте к объекту компонент Light. Реализуйте событие, чтобы при входе игрока в триггер зона свет включался, а при выходе - выключался.

6. Создайте объект с компонентом Animator Реализуйте событие, чтобы при нажатии клавиши пробел запускалась анимация прыжка.

7. Добавьте к объекту компонент Script с публичной переменной. Реализуйте событие, чтобы при изменении значения переменной происходило изменение цвета объекта.

8. Создайте систему, в которой при уничтожении одного объекта (OnDestroy) другой объект получает уведомление и выполняет действие (например, увеличивает свой размер).

9. Добавьте к объекту компонент Particle System. Реализуйте событие, чтобы при столкновении с другим объектом запускалась частица (эффект взрыва).

10. Создайте объект с компонентом UI Text. Реализуйте событие, чтобы при наведении курсора мыши на определённый объект отображалось сообщение на экране.

11. Исчезающий куб. Создайте игровой объект "Куб" (Cube) в сцене Unity. Напишите компонент (скрипт), который будет отслеживать нажатие клавиши «пробел» на клавиатуре. При нажатии этой клавиши игровой объект "Куб" должен исчезать (становиться неактивным в сцене).

12. Исчезающая сфера. Создайте игровой объект "Сфера" (Sphere) в сцене Unity. Напишите компонент (скрипт), который будет отслеживать нажатие клавиши «←» на клавиатуре. При нажатии этой клавиши игровой объект "Сфера" должен исчезать (становиться неактивным в сцене).

13. Исчезающая пирамида. Создайте игровой объект "Пирамида" (Pyramid) в сцене Unity (можно импортировать готовую модель или создать самостоятельно). Напишите компонент (скрипт), который будет отслеживать нажатие клавиши «→» на клавиатуре. При нажатии этой клавиши игровой объект "Пирамида" должен исчезать (становиться неактивным в сцене).

14. Исчезающий цилиндр. Создайте игровой объект "Цилиндр" в сцене Unity (можно импортировать готовую модель или создать самостоятельно). Напишите компонент (скрипт), который будет отслеживать нажатие клавиши «R» на клавиатуре. При нажатии этой клавиши игровой объект "Пирамида" должен исчезать (становиться неактивным в сцене).

15. Исчезающая объект. Импортируйте готовые модели понравившихся объектов. Напишите компонент (скрипт), который будет отслеживать нажатие клавиши «↑» или «↓» на клавиатуре. При нажатии одной из этих клавиш объект должен исчезать (становиться неактивным в сцене).

Задачи по теме «Создание и настройка 3D анимации».

1. Анимация движения куба. Создайте игровой объект "Куб" (Cube) в сцене. Настройте для него анимацию перемещения вперёд и назад по оси Z. Анимация должна запускаться автоматически при запуске сцены и повторяться бесконечно (циклически).

2. Вращающаяся сфера. Создайте игровой объект "Сфера" (Sphere) в сцене. Настройте для него анимацию постоянного вращения вокруг своей оси Y. Анимация должна работать непрерывно на протяжении всей сцены.

3. Анимация открытия двери. Создайте 3D-модель двери (или используйте примитивы для имитации двери). Настройте анимацию открытия двери (поворот вокруг одной из сторон). Реализуйте запуск анимации по нажатию клавиши «E» на клавиатуре.

4. Анимация изменения цвета объекта. Создайте любой 3D-объект в сцене. Настройте для него анимацию, при которой цвет материала плавно изменяется с одного цвета на другой и обратно (например, из красного в синий и обратно). Анимация должна проигрываться циклически.

5. Анимация прыжка персонажа. Создайте 3D-модель персонажа (можно использовать примитивы). Настройте анимацию прыжка: персонаж должен приседать, затем подпрыгивать вверх и возвращаться в исходное положение. Анимация должна запускаться при нажатии на клавишу «пробел».

6. Создать движение объекта по траектории, заданной функцией $f(x)=2x^2+5x-3$ на интервале $x \in [-2, 2]$.

7. Создать движение объекта по параболической траектории, заданной функцией $f(x)=2x^2+5x-3$ на интервале $x \in [-2, 2]$. Анимация должна выполняться автоматически при запуске сцены.

8. Создать анимацию движения объекта по синусоидальной траектории, заданной функцией $f(x) = 3 - \sin(\pi x)$ на интервале $x \in [0, 2]$. Объект должен двигаться вдоль оси X, а координата Y вычисляться по функции $f(x)$.

9. Реализовать движение объекта по траектории, заданной функцией $f(x) = e^x - 1$ на интервале $x \in [0, 1.5]$. Анимация должна плавно перемещать объект по этой кривой при запуске сцены.

10. Настроить движение объекта по дуге окружности радиусом 4, описываемой уравнением $x = 4 \cdot \cos(\theta)$, $y = 4 \cdot \sin(\theta)$ на интервале $\theta \in [0, \pi/2]$. Объект должен плавно проходить по дуге четверти круга.

Задачи по теме «Настройка поверхности. Визуальное оформление элемента Ground».

1. Создайте элемент Ground (земля) в виде плоскости. Примените к нему стандартный материал зеленого цвета, чтобы имитировать траву.
2. Создайте элемент Ground и наложите на него текстуру травы, чтобы поверхность выглядела более реалистично.
3. Добавьте на элемент Ground материал с текстурой песка и настройте параметры отражения, чтобы добиться эффекта сухой поверхности.
4. Создайте материал для элемента Ground, который будет плавно менять цвет с зеленого на желтый по ширине плоскости (градиент).
5. Добавьте к элементу Ground эффект нормалей (Normal Map) для создания рельефа (неровностей) на поверхности.
6. Создайте элемент Ground с текстурой камней и настройте тильный (Tile) режим повторения текстуры для равномерного покрытия большой поверхности.
7. Добавьте к элементу Ground отражающий материал, чтобы поверхность выглядела как мокрая после дождя.
8. Создайте элемент Ground и наложите на него несколько слоёв текстур (травы, грязи, камней) с помощью Splat Map или смешивания материалов.
9. Добавьте к элементу Ground детали (например, следы от колес или ног), которые отображаются поверх основной текстуры при взаимодействии с персонажем или объектами.
10. Создайте элемент Ground с анимированной текстурой (например, движущаяся вода или трава, колышущаяся на ветру).

Задачи по теме «Скрипт-файл сценария игры»

1. Создайте скрипт, который при запуске игры выводит в консоль приветственное сообщение.
2. Напишите скрипт, который двигает игровой объект вперед с постоянной скоростью при удержании клавиши «W».
3. Реализуйте скрипт, который уменьшает здоровье персонажа при столкновении с врагом. При достижении нуля - выводит сообщение «Game Over».
4. Создайте скрипт, который увеличивает счет игрока при сборе предметов (например, монет). Счет отображается на экране.
5. Напишите скрипт, который запускает анимацию прыжка при нажатии на пробел и запрещает повторный прыжок до приземления.
6. Реализуйте скрипт, который по таймеру (например, каждые 10 секунд) случайно меняет цвет игрового объекта.
7. Создайте скрипт, который отслеживает победу: если игрок собрал все ключи на уровне, появляется надпись «Победа!» и игра останавливается.
8. Напишите скрипт, который при нажатии на кнопку в интерфейсе перезапускает уровень.
9. Создайте скрипт, который сохраняет прогресс игрока (например, количество собранных монет) между сессиями.
10. Реализуйте скрипт, который управляет диалогом между персонажами: при нажатии на кнопку появляется следующая реплика.

Задачи по теме «Игровое меню. Счётчик очков»

1. Создайте главное меню с кнопками «Играть», «Настройки» и «Выход». При нажатии на каждую кнопку выполняется соответствующее действие.
2. Реализуйте отображение текущего количества очков игрока на экране с помощью UI Text или аналогичного компонента.

3. Сделайте всплывающее меню паузы, которое появляется при нажатии на клавишу Esc, и содержит кнопки «Продолжить», «Главное меню» и «Выход».

4. Напишите скрипт, который увеличивает счётчик очков на 10 при сборе специального предмета и обновляет отображение на экране.

5. Добавьте в главное меню кнопку «Таблица рекордов», при нажатии на которую открывается окно с лучшими результатами игроков.

6. Сделайте, чтобы при окончании игры (например, при проигрыше) появлялось меню с итоговым счетом и кнопками «Повторить» и «Главное меню».

7. Реализуйте анимацию увеличения счётчика очков: при получении очков число на экране плавно увеличивается до новой величины.

8. Сделайте так, чтобы при наборе определённого количества очков (например, 100) появлялось сообщение или открывалось новое меню (например, «Новый уровень!»).

9. Добавьте возможность сбросить счётчик очков до нуля через пункт меню «Сбросить прогресс».

10. Сделайте так, чтобы при начале новой игры счётчик очков автоматически сбрасывался, а интерфейс меню исчезал и появлялся игровой UI.

Описание четырёх кейсов для промежуточной аттестации (зачёт) по дисциплине. Каждый кейс включает: исходные данные, пошаговую задачу для студента, требования к реализации, критерии оценки и примерный объём кода.

«Компьютерная графика». Каждый кейс позволяет оценить достижение компетенций **ОПК-5.2**, **РЛ-3.1** и **МФ-7.3** в комплексе.

4.2 Зачетно-экзаменационные материалы для промежуточной аттестации (экзамен/зачет).

Кейс №1. Визуализация признакового пространства с выбором проекции

Условие:

Для анализа данных о клиентах интернет-магазина вы получили двумерное признаковое пространство (признаки: *средний чек* и *частота покупок*). Точки сгруппированы в три кластера. Необходимо визуализировать эти данные в игровой сцене Unity 3D таким образом, чтобы:

- кластеры были различимы,
- пользователь мог вращать камеру и приближать/отдалять объекты,
- при наведении на точку всплывала подсказка с её координатами.

Задание:

1. Обоснуйте выбор типа проекции камеры (перспективная / ортографическая) и расположения камеры относительно точек, учитывая метрические свойства данных (плотность кластеров, расстояния между ними).

2. Напишите фрагмент скрипта на C#, который реализует движение камеры за указателем мыши (вращение вокруг центра сцены).

3. Опишите, какие компоненты Unity (GameObject, Transform, Collider, UI) вы будете использовать для реализации всплывающей подсказки при наведении.

Оцениваемые индикаторы:

• **МФ-7.3** – обоснование выбора камеры и проекции в зависимости от топологии кластеров.

• **РЛ-3.1** – разработка скрипта на C# с элементами ИИ (управление камерой = адаптивное поведение).

• **ОПК-5.2** – демонстрация навыков установки и настройки компонентов камеры в Unity, конфигурации параметров проекции и управления камерой через скрипт; использование средств отладки и профилирования для сопровождения работы камеры в сцене..

Критерии оценки (зачёт / незачёт):

- *Зачёт*: обоснование логически верно (указаны причины выбора типа проекции и положения камеры), код синтаксически корректен и решает задачу (на уровне псевдокода или реального C#), перечислены необходимые компоненты Unity с пояснением их роли.
- *Незачёт*: отсутствует обоснование, код не соответствует заданию или отсутствует, не названы ключевые компоненты.

Кейс №2. Разработка простого симулятора «Сбор объектов» с адаптивным счётом

Условие:

Вам необходимо создать прототип игры для обучения агента сбору ресурсов. На сцене случайным образом появляются 10 кубов (ресурсов). Пользователь управляет сферой с помощью мыши: при касании куб исчезает, а счётчик очков увеличивается на 1. Через каждые 5 очков скорость перемещения сферы должна увеличиваться на 10% (игровая адаптация). Интерфейс должен отображать текущий счёт и сообщение о победе при сборе всех ресурсов.

Задание:

1. Напишите на C# скрипт для сферы, который реализует:
 - перемещение за указателем мыши (по плоскости),
 - обработку столкновения (OnCollisionEnter) с кубом (уничтожение куба, увеличение счёта),
 - изменение скорости перемещения при достижении порога очков.
2. Опишите, как вы организуете спавн кубов в случайных позициях (достаточно алгоритма).
3. Обоснуйте, почему данный прототип можно считать «прикладным решением с элементами ИИ» (п. 2 задания).

Оцениваемые индикаторы:

- **PL-3.1** – разработка и отладка прикладного решения на C# с элементами адаптации (скорость растёт).
- **ОПК-5.2** – демонстрация умения настраивать игровое окружение (спавн объектов, теги, UI), устанавливать и сопровождать скриптовую логику (обработка коллизий, адаптивное изменение скорости), проводить отладку в Unity.
- **MF-7.3** – неявно (визуализация пространства ресурсов и их распределение – «признаковое пространство» с равномерной метрикой); обоснование выбора способа спавна (случайное равномерное или с учётом плотности) может быть дополнительным баллом.

Критерии:

- *Зачёт*: корректная структура скрипта (Start, Update, OnCollisionEnter), реализован счёт и изменение скорости, описан простой алгоритм спавна, дано аргументированное объяснение «элементов ИИ».
- *Незачёт*: скрипт не компилируется или содержит грубые логические ошибки, отсутствует объяснение, не описана механика спавна.

Кейс №3. Настройка визуализации поверхности на основе данных о высоте

Условие:

У вас есть массив высот (100×100 точек) для участка местности. Данные получены с датчика и содержат шум. Вам необходимо визуализировать рельеф в Unity так, чтобы:

- хорошо читались локальные перепады высот (холмы и впадины),
- можно было быстро определить зоны с аномально большим уклоном,
- визуализация не вводила в заблуждение относительно метрических свойств (расстояний между точками).

Задание:

1. Предложите метод создания поверхности (Mesh, Terrain, Plane с модификацией вершин). Обоснуйте выбор с точки зрения топологии (связность вершин) и метрик (искажение расстояний при визуализации).

2. Опишите, как вы примените цвета или текстуры (шейдер) для выделения зон с большим уклоном.

3. Напишите псевдокод функции, которая по массиву высот создаёт сетку (Mesh) с заданным разрешением.

Оцениваемые индикаторы:

- **MF-7.3** – обоснование выбора метода визуализации/трансформации высот (Mesh vs Terrain) с учётом топологических (связность) и метрических (сохранение расстояний) свойств.

- **ОПК-5.2** – настройка процедурного создания Mesh как части графической системы Unity; установка параметров геометрии (вершины, индексы, нормали) и сопровождение визуализации рельефа с помощью профилирования производительности (оценка количества полигонов, оптимизация отрисовки)..

- **PL-3.1** – предлагаемый подход (псевдокод) может быть реализован на C# в Unity.

Критерии:

- *Зачёт*: выбор обоснован (например, Mesh даёт полный контроль, Terrain быстрее для больших ландшафтов), предложен разумный способ раскраски (градиент по наклону, нормали), псевдокод корректен (циклы по x, z, создание вершин и треугольников).

- *Незачёт*: нет обоснования, псевдокод отсутствует или содержит фатальные ошибки, способ раскраски не описан или не связан с уклоном.

Кейс №4 (МИП). Физически корректный и оптимизированный расчёт освещения на основе карты нормалей

Основан на технологическом стеке компании CarX TECHNOLOGIES.

Условие:

Реализовать PBR-модель освещения с использованием карты нормалей в Unity URP. Требуется достичь максимальной производительности и приближения к реальному поведению света.

Задание (пошагово):

1. Обосновать выбор пространства вычислений (Tangent Space / World Space) для нормалей.

2. Спроектировать структуры `v2f` и `appdata` с минимальным набором атрибутов (позиция, нормаль, тангент, битангент, UV).

3. Написать вершинный шейдер: передача позиций, построение TBN-матрицы.

4. Реализовать фрагментный шейдер: сэмплирование и распаковка карты нормалей (`UnpackNormal`), преобразование векторов в нужное пространство.

5. Реализовать диффузную (Ламберт) и specularную (упрощённая GGX или Блинн-Фонг) составляющие с учётом затухания света и $N \cdot L$.

6. Применить оптимизации: половинная точность (`half`) для векторов освещения, замена `pow`, `sqrt` на аппроксимации, минимизация текстурных выборов.

7. Протестировать на разных типах поверхностей (плоские, изогнутые, с высокой детализацией карт нормалей). Сравнить с Lit-шейдером Unity по качеству и производительности (Frame Debugger, Profiler).

Оцениваемые индикаторы: ОПК-5.2 настройка рендер-пайплайна URP, установка и конфигурация шейдера, использование Frame Debugger для диагностики графического конвейера, PL-3.1, MF-7.3.

Критерии зачёта:

- *Зачёт*: рабочий `.shader`-файл, отсутствие визуальных артефактов, чистый модульный код с комментариями, обоснование архитектурных выборов (пространство вычислений, оптимизации).

- *Незачёт*: шейдер не компилируется, наличие артефактов, отсутствие обоснования оптимизаций.

Время подготовки: 20–30 минут. **Форма ответа:** устное изложение с демонстрацией фрагментов кода.

Общие требования ко всем кейсам для промежуточной аттестации

Преподаватель может предложить студенту:

- **Один из четырех кейсов** (выбирает случайно или по билету).
- **Время подготовки** – 20–30 минут.
- **Форма ответа** – устное изложение с демонстрацией фрагментов кода (на бумаге или ноутбуке).

- **Итоговая оценка** – «зачёт»/«незачёт» по совокупности критериев выбранного кейса.

Кейсы покрывают все индикаторы:

- MF-7.3 – кейсы 1 и 3.
- PL-3.1 и ОПК-5.2 – кейсы 1 (скрипт камеры), 2 (скрипт сферы), 3 (псевдокод mesh).

Критерии оценивания результатов обучения

Критерии оценки лабораторной работы

При защите лабораторной работы оцениваются следующие аспекты:

Критерий	Вес (баллы)	Описание
Полнота выполнения задания	25	Все пункты задания реализованы в полном объеме, без пропусков.
Работоспособность	25	Разработанный проект / скрипт корректно работает в Unity (нет ошибок выполнения, логических сбоев, вылетов).
Качество кода (для программируемых работ)	20	Соблюдение стандартов оформления (именование переменных, отступы, комментарии), отсутствие дублирования, обработка граничных условий.
Понимание материала	20	Студент может объяснить выбранные подходы, ответить на вопросы о работе компонентов, обосновать принятые решения.
Оформление отчёта	10	Наличие скриншотов, листингов, описания алгоритмов и результатов тестирования.

Итого: 100 баллов.

Оценка «зачтено» – 55 баллов и выше, **«не зачтено»** – менее 55 баллов.

Индикаторно-ориентированные критерии (по компетенциям)

Дополнительные баллы для точной оценки соответствия компетенциям можно использовать следующие дополнительные критерии:

ОПК-5.2 (реализация и интеграция кода)

- **Максимум (10 баллов)** – продемонстрирована полная установка и настройка графического решения (настройка рендер-пайплайна, импорт ассетов, конфигурация материалов/шейдеров), проведена диагностика через Frame Debugger/Profiler, обеспечено техническое сопровождение (комментарии, логирование, обработка ошибок).

- **Минимум** – код работает «как есть», но не читаем, нет обработки ошибок.

PL-3.1 (прикладное решение с элементами ИИ)

- **Максимум (10 баллов)** – в работе присутствуют адаптивные алгоритмы (изменение скорости, счётчик, "умное" перемещение), студент может объяснить их назначение.

- **Минимум** – решение линейное, без адаптации.

MF-7.3 (обоснование выбора визуализации)

- **Максимум (10 баллов)** – студент аргументированно объясняет, почему выбрана именно такая камера, проекция, материал или способ отображения данных, ссылаясь на топологические и метрические свойства.

- **Минимум** – выбор случаен или не обоснован.

Эти дополнительные баллы могут добавляться к основным 100.

Примеры применения критериев к конкретным лабораторным работам
Лабораторная работа №7 «Скрипт-файл сценария игры» (перемещение, коллизии, разрушение)

Критерий	Макс. балл	Типичные требования
Полнота	25	Реализованы все три механики: перемещение за мышью, столкновение и разрушение объекта, обновление счётчика (если требуется).
Работоспособность	25	При запуске сцены объект следует за курсором, при касании куба – куб исчезает, счёт увеличивается. Нет ошибок в консоли.
Качество кода	20	Код разбит на методы (Start, Update, OnCollisionEnter), переменные названы осмысленно, есть комментарии.
Понимание	20	Студент объясняет, зачем нужен Rigidbody, почему используется OnCollisionEnter, а не триггер.
Оформление	10	Приложен скриншот сцены и листинг скрипта.

Лабораторная работа №5 «Настройка поверхности. Визуальное оформление Ground»

Критерий	Макс. балл	Типичные требования
Полнота	25	Создана поверхность, применена текстура, настроен материал (цвет, отражение, нормали).
Работоспособность	25	Текстура корректно отображается, нет артефактов, тилинг работает.
Обоснование (MF-7.3)	20	Студент объясняет, почему выбрана именно эта текстура/шейдер для демонстрации высот или уклона.
Понимание	20	Отвечает на вопросы о параметрах нормалей, Tile, Metallic.
Оформление	10	Скриншоты до/после, описание настроек.

Рекомендации по проведению защиты

- **Формат защиты:** индивидуальное собеседование (5–10 минут) или демонстрация работающего проекта.
- **Обязательные вопросы:**
 - Какие компоненты Unity использованы?
 - Какой код отвечает за ключевую механику?
 - Почему выбран именно такой метод визуализации (для MF-7.3)?
 - Как вы тестировали работоспособность?
- **Фиксация оценки:** ведомость с оценкой «зачёт» / «незачёт».

Текущий контроль осуществляется в течение семестра по защите лабораторных работ (не менее 6 из 8 защищённых ЛР), каждая с оценкой не ниже 55 баллов и выполнению типовых задач (не менее 10 задач в электронном журнале).

Промежуточная аттестация – зачёт, проводится в устной форме по кейсам. Время на подготовку – 30 минут, ответ – до 15 минут. **Оценка «зачёт»** ставится при успешном выполнении не менее 2 из 3 заданий кейса (для кейса №2 – обязательно наличие рабочего фрагмента кода; для кейса №1 и №3 – развёрнутое обоснование и корректный код/псевдокод, для кейса №4 – особые условия).

4.3 Методические указания по организации лабораторных работ по дисциплине «Компьютерная графика»

Методические указания по выполнению лабораторной работы №6 «Скрипт-файлы сценария игры»

1. Цель работы

Сформировать практические навыки разработки скриптов на языке C# в среде Unity для управления игровыми объектами, обработки столкновений, реализации логики разрушения объектов и перемещения объекта за указателем мыши.

2. Задачи работы

- Создать скрипт, обеспечивающий перемещение игрового объекта (сферы) по плоскости XZ за движением указателя мыши.
- Реализовать обработку столкновения (OnCollisionEnter) с объектом «кубом», в результате чего куб уничтожается.
- Добавить счётчик очков, увеличивающийся при уничтожении каждого куба.
- Организовать появление сообщения о победе при уничтожении всех кубов.

3. Требуемые компетенции (результаты обучения)

- **ОПК-5.2** – способность устанавливать, настраивать и сопровождать программное обеспечение: реализация скриптов на C# в Unity, конфигурация компонентов физики (Rigidbody, Collider), настройка UI и системы событий, диагностика работы с помощью Profiler.
- **PL-3.1** – реализация прикладного решения с элементами адаптации (счётчик очков, изменение скорости или поведения при достижении порога).

4. Теоретические сведения (кратко)

- **Transform** – компонент, определяющий положение, поворот и масштаб объекта.
- **Rigidbody** – компонент физического тела, необходим для корректной обработки коллизий и движения под действием физики (даже если движение управляется скриптом).
- **Collider** – определяет форму объекта для столкновений.
- **OnCollisionEnter(Collision collision)** – метод MonoBehaviour, вызываемый при касании двух объектов, имеющих Collider и Rigidbody.
- **Destroy(GameObject)** – удаляет объект со сцены.
- **Векторное преобразование экранных координат в мировые** – использование Camera.main.ScreenToWorldPoint для перемещения объекта под курсором мыши.

5. Необходимое программное обеспечение и оборудование

- Unity Hub + Unity Editor (версия 2021 LTS или новее).
- Visual Studio / Rider с установленной нагрузкой для разработки на C# и Unity.
- Компьютер с ОС Windows 10/11 (или macOS), видеокартой с поддержкой DirectX

11.

6. Порядок выполнения работы

Шаг 1. Создание сцены и базовых объектов

1. Открыть Unity, создать новый 3D проект (или использовать ранее созданный).
2. В сцене разместить:
 - **Плоскость (Ground)** – Plane (GameObject → 3D Object → Plane). Переименовать в «Ground».
 - **Игровой объект – сфера (Player)** – Sphere. Переименовать в «Player», задать Position (0, 0.5, 0).
 - **Несколько объектов-кубов (Targets)** – Cube. Создать 5–10 кубов, разместить в случайных позициях на плоскости (можно вручную).
3. Добавить компоненты:
 - К сфере «Player»: **Rigidbody** (масса = 1, Use Gravity = true), **Sphere Collider** (по умолчанию).
 - К каждому кубу: **Box Collider**, при необходимости **Rigidbody** (но можно без Rigidbody, если куб не должен падать; для срабатывания OnCollisionEnter достаточно иметь Rigidbody хотя бы у одного из сталкивающихся объектов, лучше у игрока).

Шаг 2. Создание скрипта для управления игроком

1. В окне Project создать папку «Scripts». Внутри создать C# скрипт «PlayerController».
2. Открыть скрипт в редакторе и заменить содержимое на код (листинг 1).

3. Прикрепить скрипт к объекту «Player».

Листинг 1 – PlayerController.cs

C#

```
using UnityEngine;
using UnityEngine.UI; // для работы с UI Text

public class PlayerController : MonoBehaviour
{
    [Header("Настройки движения")]
    public float speed = 5f;      // скорость перемещения
    private Rigidbody rb;

    [Header("Игровые параметры")]
    public int score = 0;         // текущий счёт
    public Text scoreText;       // ссылка на текстовое поле UI
    public Text winText;         // ссылка на текстовое поле победы
    private int totalTargets;     // общее количество кубов

    void Start()
    {
        rb = GetComponent<Rigidbody>();
        // Подсчитываем все объекты с тегом "Target" (не забудьте назначить тег кубам)
        totalTargets = GameObject.FindGameObjectsWithTag("Target").Length;
        UpdateScoreUI();
        if (winText != null) winText.gameObject.SetActive(false);
    }

    void FixedUpdate()
    {
        // Движение за указателем мыши по плоскости XZ
        Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);
        RaycastHit hit;
        if (Physics.Raycast(ray, out hit, 100f))
        {
            Vector3 targetPoint = hit.point;
            targetPoint.y = transform.position.y; // сохраняем высоту сферы
            Vector3 moveDirection = (targetPoint - transform.position).normalized;
            Vector3 newPosition = transform.position + moveDirection * speed *
Time.fixedDeltaTime;
            rb.MovePosition(newPosition);
        }
    }

    void OnCollisionEnter(Collision collision)
    {
        // При столкновении с объектом, имеющим тег "Target"
        if (collision.gameObject.CompareTag("Target"))
        {
            Destroy(collision.gameObject);
            score++;
        }
    }
}
```

```

        UpdateScoreUI();

        if (score >= totalTargets)
        {
            Win();
        }
    }
}

void UpdateScoreUI()
{
    if (scoreText != null)
        scoreText.text = "Счёт: " + score.ToString();
}

void Win()
{
    if (winText != null)
    {
        winText.gameObject.SetActive(true);
        winText.text = "Победа! Все объекты собраны!";
    }
    // Остановить движение (опционально)
    speed = 0f;
}
}

```

Шаг 3. Настройка тегов и UI

1. Выделить все кубы, в инспекторе в выпадающем списке **Tag** выбрать «Add Tag...», создать тег «Target», затем назначить его каждому кубу.
2. Создать элементы интерфейса:
 - **Canvas** (GameObject → UI → Canvas).
 - На Canvas создать **Text** (правой кнопкой → UI → Text – Legacy). Переименовать в «ScoreText». Расположить в левом верхнем углу.
 - Создать ещё один Text – «WinText», разместить по центру, сделать большим шрифтом, отключить галочку «Active» (чтобы изначально было скрыто).
3. В скрипте **PlayerController** перетащить созданные объекты Text в поля **Score Text** и **Win Text** в инспекторе.

Шаг 4. Настройка камеры

Для удобства переместить камеру так, чтобы она смотрела на сцену сверху под углом (Position: (0, 10, -10), Rotation: (45, 0, 0)).

Шаг 5. Тестирование

1. Нажать **Play**.
2. Проверить:
 - Сфера движется за курсором мыши по плоскости.
 - При касании куба – куб исчезает, счёт увеличивается.
 - При сборе всех кубов появляется сообщение о победе.
3. Открыть консоль (Window → General → Console) – убедиться в отсутствии ошибок.

Шаг 6. Дополнительное задание (для повышенной оценки)

- Реализовать увеличение скорости сферы на 10% после каждого 5 собранных кубов.
- Добавить звук уничтожения куба (AudioSource.PlayClipAtPoint).

- Добавить анимацию вращения кубов перед исчезновением.

7. Содержание отчёта по лабораторной работе

Отчёт должен содержать:

1. **Титульный лист** (название работы, ФИО, группа).
2. **Цель и задачи** (из пунктов 1,2 методички).
3. **Краткие теоретические сведения** (основные компоненты события OnCollisionEnter, Destroy, перемещение мыши).
4. **Листинг разработанного скрипта** (с комментариями).
5. **Скриншоты сцены до и после сбора объектов** (2–3 изображения).
6. **Результаты тестирования** (описание, всё ли работает, выявленные ошибки и их исправление).
7. **Ответы на контрольные вопросы.**
8. **Вывод** (что сделано, какие компетенции освоены).
8. **Контрольные вопросы для защиты**
 1. Почему в скрипте используется FixedUpdate(), а не Update()? *(Для работы с физикой, чтобы движение было синхронизировано с шагами физического движка.)*
 2. Как работает преобразование координат мыши в мировые координаты через ScreenPointToRay?
 3. Зачем сфере нужен компонент Rigidbody, если её перемещение управляется скриптом?
 4. Почему условие if (collision.gameObject.CompareTag("Target")) безопаснее, чем сравнение по имени?
 5. Как можно добавить эффект «взрыва» при уничтожении куба?
 6. Каким образом реализовать увеличение скорости при наборе определённого количества очков?

9. Критерии оценки выполнения работы (в баллах)

Критерий	Макс. балл
Полнота реализации (движение + столкновение + уничтожение + счёт + победа)	25
Работоспособность (нет ошибок, все механики корректны)	25
Качество кода (комментарии, именование, обработка граничных случаев)	20
Понимание материала (устные ответы на защите)	20
Оформление отчёта (скриншоты, листинг, вывод)	10
Итого	100

Для зачёта необходимо набрать не менее 55 баллов.

10. Рекомендуемая литература для подготовки

- Unity Documentation: MonoBehaviour.OnCollisionEnter (<https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnCollisionEnter.html>)
- Unity Manual: Controlling Game Objects with Scripts (<https://learn.unity.com/tutorial/controlling-game-objects-with-scripts>)

Примечание: Данный пример может быть адаптирован для любой другой лабораторной работы (анимация, настройка поверхности, камера и т.д.) путём замены темы, листинга и контрольных вопросов.

Методические указания по выполнению лабораторных работ индустриального партнёра (МИП) – разрабатываются отдельно и размещаются в электронной информационно-образовательной среде. Ниже приводится **общая структура** для одной из них (полные версии – в Moodle).

Пример: Лабораторная работа №8 (МИП) «Проектирование и реализация базового пользовательского шейдера»

Цель: освоить создание шейдера с TBN-матрицей и картой нормалей.

Задачи:

- Написать вершинный и фрагментный шейдеры.
- Построить TBN-матрицу.
- Реализовать диффузное освещение.

Теоретические сведения: этапы графического конвейера, касательное пространство, упаковка нормалей.

Порядок выполнения (2 часа):

1. Создать шейдерный файл в Unity (ShaderLab/HLSL).
2. Определить структуры `appdata` и `v2f`.
3. В вершинном шейдере вычислить TBN-матрицу.
4. Во фрагментном шейдере сэмплировать карту нормалей, распаковать `UnpackNormal`.
5. Реализовать простое освещение.
6. Протестировать на модели с нормал-мапой, проанализировать через Frame Debugger.

Контрольные вопросы:

- Почему касательное пространство предпочтительнее мирового?
- Как рассчитывается битангент?
- Зачем нужна нормализация векторов во фрагментном шейдере?

Критерии оценки (100 баллов): полнота реализации (25), работоспособность (25), **качество реализации алгоритмов и использования структур данных (например, TBN-матрица, передача атрибутов)** (20), понимание материала (20), оформление отчёта (10). Зачёт -- 55 баллов и выше.

Оценочные средства для инвалидов и лиц с ограниченными возможностями здоровья выбираются с учетом их индивидуальных психофизических особенностей.

- при необходимости инвалидам и лицам с ограниченными возможностями здоровья предоставляется дополнительное время для подготовки ответа на экзамене;
- при проведении процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья предусматривается использование технических средств, необходимых им в связи с их индивидуальными особенностями;
- при необходимости для обучающихся с ограниченными возможностями здоровья и инвалидов процедура оценивания результатов обучения по дисциплине может проводиться в несколько этапов.

Процедура оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья по дисциплине (модулю) предусматривает предоставление информации в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа.

Для лиц с нарушениями слуха:

- в печатной форме,

– в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

– в печатной форме,

– в форме электронного документа.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

5. Перечень учебной литературы, информационных ресурсов и технологий

5.1. Учебная литература

1. Гаркуша, О.В. Unity в картинках: учебное пособие - Краснодар: Кубанский гос. ун-т, 2024. - 167 с.

2. Воронина, В. В. Компьютерная графика : учебное пособие / В. В. Воронина, В. В. Шишкин. — Ульяновск : Ульяновский государственный технический университет, 2023. — 176 с. — ISBN 978-5-9795-2328-6. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/149269.html>

3. Кенни Ламмерс. «Шейдеры и эффекты в Unity. Книга рецептов» (2-е изд.). Издательство: ДМК Пресс, 2023.

5.2. Периодическая литература

1. Базы данных компании «Ист Вью» <http://dlib.eastview.com>

2. Электронная библиотека GREBENNIKON.RU <https://grebennikon.ru/>

5.3. Интернет-ресурсы, в том числе современные профессиональные базы данных и информационные справочные системы

Электронно-библиотечные системы (ЭБС):

1. ЭБС «ЮРАЙТ» <https://urait.ru/>

2. ЭБС «УНИВЕРСИТЕТСКАЯ БИБЛИОТЕКА ОНЛАЙН» www.biblioclub.ru

3. ЭБС «BOOK.ru» <https://www.book.ru>

4. ЭБС «ZNANIUM.COM» www.znanium.com

5. ЭБС «ЛАНЬ» <https://e.lanbook.com>

Профессиональные базы данных:

1. Web of Science (WoS) <http://webofscience.com/>

2. Scopus <http://www.scopus.com/>

3. ScienceDirect www.sciencedirect.com

4. Журналы издательства Wiley <https://onlinelibrary.wiley.com/>

5. Научная электронная библиотека (НЭБ) <http://www.elibrary.ru/>

6. Полнотекстовые архивы ведущих западных научных журналов на Российской платформе научных журналов НЭИКОН <http://archive.neicon.ru>

7. Национальная электронная библиотека (доступ к Электронной библиотеке диссертаций Российской государственной библиотеки (РГБ) <https://rusneb.ru/>

8. Президентская библиотека им. Б.Н. Ельцина <https://www.prlib.ru/>

9. Электронная коллекция Оксфордского Российского Фонда

<https://ebookcentral.proquest.com/lib/kubanstate/home.action>

10. Springer Journals <https://link.springer.com/>

11. Nature Journals <https://www.nature.com/siteindex/index.html>

12. Springer Nature Protocols and Methods

<https://experiments.springernature.com/sources/springer-protocols>

13. Springer Materials <http://materials.springer.com/>

14. zbMath <https://zbmath.org/>

15. Nano Database <https://nano.nature.com/>

16. Springer eBooks: <https://link.springer.com/>

17. "Лекториум ТВ" <http://www.lektorium.tv/>
18. Университетская информационная система РОССИЯ <http://uisrussia.msu.ru>

Информационные справочные системы:

1. Консультант Плюс - справочная правовая система (доступ по локальной сети с компьютеров библиотеки)

Ресурсы свободного доступа:

1. Американская патентная база данных <http://www.uspto.gov/patft/>
2. Полные тексты канадских диссертаций <http://www.nlc-bnc.ca/thesescanada/>
3. КиберЛенинка (<http://cyberleninka.ru/>);
4. Министерство науки и высшего образования Российской Федерации <https://www.minobrnauki.gov.ru/>;
5. Федеральный портал "Российское образование" <http://www.edu.ru/>;
6. Информационная система "Единое окно доступа к образовательным ресурсам" <http://window.edu.ru/>;
7. Единая коллекция цифровых образовательных ресурсов <http://school-collection.edu.ru/> .
8. Федеральный центр информационно-образовательных ресурсов (<http://fcior.edu.ru/>);
9. Проект Государственного института русского языка имени А.С. Пушкина "Образование на русском" <https://pushkininstitute.ru/>;
10. Справочно-информационный портал "Русский язык" <http://gramota.ru/>;
11. Служба тематических толковых словарей <http://www.glossary.ru/>;
12. Словари и энциклопедии <http://dic.academic.ru/>;
13. Образовательный портал "Учеба" <http://www.ucheba.com/>;
14. Законопроект "Об образовании в Российской Федерации". Вопросы и ответы http://xn--273--84d1f.xn--p1ai/voprosy_i_otvety

Собственные электронные образовательные и информационные ресурсы КубГУ:

1. Среда модульного динамического обучения <http://moodle.kubsu.ru>
2. База учебных планов, учебно-методических комплексов, публикаций и конференций <http://mschool.kubsu.ru/>
3. Библиотека информационных ресурсов кафедры информационных образовательных технологий <http://mschool.kubsu.ru/>;
4. Электронный архив документов КубГУ <http://docspace.kubsu.ru/>
5. Электронные образовательные ресурсы кафедры информационных систем и технологий в образовании КубГУ и научно-методического журнала "ШКОЛЬНЫЕ ГОДЫ" <http://icdau.kubsu.ru/>

6. Методические указания для обучающихся по освоению дисциплины (модуля)

В освоении дисциплины инвалидами и лицами с ограниченными возможностями здоровья большое значение имеет индивидуальная учебная работа (консультации) – дополнительное разъяснение учебного материала.

Индивидуальные консультации по предмету являются важным фактором, способствующим индивидуализации обучения и установлению воспитательного контакта между преподавателем и обучающимся инвалидом или лицом с ограниченными возможностями здоровья.

По курсу предусмотрено проведение лекционных занятий, на которых дается систематизированный материал. В ходе лекций рассматриваются ключевые концепции.

Лабораторные занятия курса посвящены практическому освоению предложенных технологий и подходов.

При самостоятельной работе студентам необходимо изучать рекомендованную литературу в виде официальной документации к используемым открытым программным продуктам, облачным платформам.

Важнейшим компонентом курса является самостоятельная проектная работа, в ходе которой студент разрабатывает законченное решение для решения задач кейсов.

7. Материально-техническое обеспечение по дисциплине (модулю)

Виртуальные машины, кластер Managed Kubernetes и ресурсы GPU в облаке предоставляется индустриальным партнером ПАО «Сбербанк»:

№	Продукт	Параметры продукта	Кол-во	Кол-во конфигураций	Ед. изм.
1	Виртуальная машина	Виртуальная машина 10% vCPU 2 vCPU 4 RAM	1	60	Шт
		ОС Ubuntu 22.04	1		Шт
		Системный диск SSD	1		Шт
			10		Гб
		Аренда публичного IP	1		Шт
2	Виртуальная машина с GPU	Виртуальная машина с GPU NVIDIA® Tesla® V100 2 GPU 8 vCPU 128 Гб RAM	1	1	Шт
		ОС Ubuntu_24.04	1		Шт
		Системный диск SSD	1		Шт
			2000		Гб
		Диск SSD	1		Шт
			4096		Гб
		Диск SSD	1		Шт
			4096		Гб
		Аренда публичного IP	1		Шт
3	K8S	Master node 8 vCPU 16 RAM	1	1	Шт
		Worker node 10% доля 4 vCPU 32 RAM	5		Шт
		Worker node SSD-NVME	64		Гб
		Аренда публичного IP	1		Шт
4	GPU	Время работы в месяц	40	1	Ч
		Инстанс 8 x NVIDIA® H100 NVLink PCIe 160 vCPU 1520 GB RAM	1		Шт
		Количество запросов к ML-моделям	1		Млн. Шт
		Кэш ML-моделей	160		Гб
5	LLM	Токены GigaChat 2 Max	50		Млн. Шт

		Токены Embeddings	400		Млн. Шт
--	--	-------------------	-----	--	------------

Наименование специальных помещений	Оснащенность специальных помещений	Перечень лицензионного программного обеспечения
Учебные аудитории для проведения занятий лекционного типа	Мебель: учебная мебель Технические средства обучения: экран, проектор, компьютер ауд. 129, 131, А-305, А-307	MS Office Word 2016 и выше Ms Power Point 2016 и выше
Учебные аудитории для проведения текущего контроля (Ауд. 101, 102, 105/1, 106 и 106а)	Мебель: учебная мебель Технические средства обучения: Экран, компьютер Оборудование: компьютерная техника с подключением к информационно-коммуникационной сети «Интернет» и доступом в электронную информационно-образовательную среду образовательной организации	Браузер Google Chrome, Jupyter Notebook 6.3.0 и выше (язык Python с библиотеками Numpy, Pandas, gensim, NLTK, PyMorphy, фреймворком PyTorch)
Учебные аудитории для проведения промежуточной аттестации (Ауд. 129, 131, А-305, А-307)	Мебель: учебная мебель	-
Учебные аудитории для проведения лабораторных работ (Ауд. 101, 102, 105/1, 106 и 106а)	Мебель: учебная мебель Технические средства обучения: экран, компьютер Оборудование: компьютерная техника с подключением к информационно-коммуникационной сети «Интернет» и доступом в электронную информационно-образовательную среду образовательной организации	Браузер Google Chrome, Jupyter Notebook 6.3.0 и выше (язык Python с библиотеками Numpy, Pandas, gensim, NLTK, PyMorphy, фреймворком PyTorch)

Для самостоятельной работы обучающихся предусмотрены помещения, укомплектованные специализированной мебелью, оснащенные компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду университета.

Наименование помещений для самостоятельной работы обучающихся	Оснащенность помещений для самостоятельной работы обучающихся	Перечень лицензионного программного обеспечения
Помещение для самостоятельной работы обучающихся (читальный зал Научной библиотеки)	Мебель: учебная мебель Комплект специализированной мебели: компьютерные столы Оборудование: компьютерная техника с подключением к информационно-коммуникационной сети «Интернет» и доступом в электронную информационно-образовательную среду образовательной организации, веб-камеры, коммуникационное оборудование, обеспечивающее доступ к сети интернет (проводное соединение и беспроводное соединение по технологии Wi-Fi)	MS Office Word 2016 и выше Ms Power Point 2016 и выше
Помещение для самостоятельной работы обучающихся (Ауд. 101, 102, 103, 105/1, 106 и 106а)	Мебель: учебная мебель Комплект специализированной мебели: компьютерные столы Оборудование: компьютерная техника с подключением к информационно-коммуникационной сети «Интернет» и доступом	Браузер Google Chrome, Matlab, Jupyter Notebook 6.3.0 и выше (язык Python с библиотеками Numpy, Pandas, gensim,

	в электронную информационно-образовательную среду образовательной организации, веб-камеры, коммуникационное оборудование, обеспечивающее доступ к сети интернет (проводное соединение и беспроводное соединение по технологии Wi-Fi)	NLTK, PyMorphu, фреймворком PyTorch)
--	--	--------------------------------------

В освоении дисциплины инвалидами и лицами с ограниченными возможностями здоровья большое значение имеет индивидуальная учебная работа (консультации) – дополнительное разъяснение учебного материала.

Индивидуальные консультации по предмету являются важным фактором, способствующим индивидуализации обучения и установлению воспитательного контакта между преподавателем и обучающимся инвалидом или лицом с ограниченными возможностями здоровья.

Примечание: Конкретизация аудиторий и их оснащение определяется ОПОП.