



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Магнитогорский государственный технический университет им. Г.И.  
Носова»



УТВЕРЖДАЮ  
Директор ИЭиАС  
В.Р. Храмшин

03.02.2026 г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)**

***ОСНОВЫ РАЗРАБОТКИ НА UNITY 3D***

Направление подготовки (специальность)  
09.03.03 Прикладная информатика

Направленность (профиль/специализация) программы  
Разработка компьютерных игр и AR/VR-приложений (виртуальной/дополненной  
реальности)

Уровень высшего образования - бакалавриат

Форма обучения  
очная

Институт/ факультет	Институт энергетики и автоматизированных систем
Кафедра	Бизнес-информатики и информационных технологий
Курс	2
Семестр	3

Магнитогорск  
2026 год

Рабочая программа составлена на основе ФГОС ВО - бакалавриат по направлению подготовки 09.03.03 Прикладная информатика (приказ Минобрнауки России от 19.09.2017 г. № 922)


Рабочая программа рассмотрена и одобрена на заседании кафедры Бизнес-информатики и информационных технологий  
21.01.2026, протокол № 5

Зав. кафедрой  Г.Н. Чусавитина

Рабочая программа одобрена методической комиссией ИЭиАС  
03.02.2026 г. протокол № 5

Председатель  В.Р. Храпшин

Рабочая программа составлена:

доцент кафедры кафедры БИиИТ, канд. пед. наук  Л.В. Курзаевой

Рецензент:

главный специалист службы бизнес-решений  
ЗАО «КОНСОМ СКС», канд. техн. наук



В.А. Ошурков

## Лист актуализации рабочей программы

---

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2027 - 2028 учебном году на заседании кафедры Бизнес-информатики и информационных

Протокол от \_\_\_\_\_ 20\_\_ г. № \_\_\_\_  
Зав. кафедрой \_\_\_\_\_ Г.Н. Чусавитина

---

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2028 - 2029 учебном году на заседании кафедры Бизнес-информатики и информационных

Протокол от \_\_\_\_\_ 20\_\_ г. № \_\_\_\_  
Зав. кафедрой \_\_\_\_\_ Г.Н. Чусавитина

---

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2029 - 2030 учебном году на заседании кафедры Бизнес-информатики и информационных

Протокол от \_\_\_\_\_ 20\_\_ г. № \_\_\_\_  
Зав. кафедрой \_\_\_\_\_ Г.Н. Чусавитина

---

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2030 - 2031 учебном году на заседании кафедры Бизнес-информатики и информационных

Протокол от \_\_\_\_\_ 20\_\_ г. № \_\_\_\_  
Зав. кафедрой \_\_\_\_\_ Г.Н. Чусавитина

### **1 Цели освоения дисциплины (модуля)**

Целью освоения дисциплины является овладение практическими знаниями, умениями и навыками в области разработки приложений на игровом движке Unity

### **2 Место дисциплины (модуля) в структуре образовательной программы**

Дисциплина Основы разработки на Unity 3D входит в часть учебного плана формируемую участниками образовательных отношений образовательной программы.

Для изучения дисциплины необходимы знания (умения, владения), сформированные в результате изучения дисциплин/ практик:

Объектно-ориентированное программирование на C#

Знания (умения, владения), полученные при изучении данной дисциплины будут необходимы для изучения дисциплин/практик:

Разработка компьютерных игр и мультимедийных приложений

### **3 Компетенции обучающегося, формируемые в результате освоения дисциплины (модуля) и планируемые результаты обучения**

В результате освоения дисциплины (модуля) «Основы разработки на Unity 3D» обучающийся должен обладать следующими компетенциями:

Код индикатора	Индикатор достижения компетенции
ОПК-7	Способен разрабатывать алгоритмы и программы, пригодные для практического применения;
ОПК-7.1	Определяет средства разработки программных средств для решения практических задач профессиональной деятельности
ОПК-7.2	Разрабатывает алгоритмы и программы для решения прикладных задач различных классов

#### 4. Структура, объём и содержание дисциплины (модуля)

Общая трудоемкость дисциплины составляет 3 зачетных единиц 108 акад. часов, в том числе:

- контактная работа – 55 акад. часов;
- аудиторная – 54 акад. часов;
- внеаудиторная – 1 акад. часов;
- самостоятельная работа – 53 акад. часов;
- в форме практической подготовки – 0 акад. час;

Форма аттестации - зачет

Раздел/ тема дисциплины	Семестр	Аудиторная контактная работа (в акад. часах)			Самостоятельная работа студента	Вид самостоятельной работы	Форма текущего контроля успеваемости и промежуточной аттестации	Код компетенции
		Лек.	лаб. зан.	практ. зан.				
1. Основы работы в игровом движке								
1.1 Введение в Unity и интерфейс	3	2	2/2И			Самостоятельная проработка темы	Опрос по принципам работы	ОПК-7.1, ОПК-7.2
1.2 Основы программирования на C# для Unity		2	6/2И		10	Самостоятельная проработка темы	Опрос по принципам работы движка	ОПК-7.1, ОПК-7.2
1.3 Физика и взаимодействие объектов		2	4/2И		10	Самостоятельная проработка темы	Опрос по принципам работы	ОПК-7.1, ОПК-7.2
Итого по разделу		6	12/6И		20			
2. Механики и визуализация								
2.1 Ввод (Input System) и движение	3	2	6		10	Самостоятельная проработка темы	Опрос по принципам работы	ОПК-7.1, ОПК-7.2
2.2 Префабы и управление ресурсами		2	4/0,2И			Самостоятельная проработка темы	Опрос по принципам работы	ОПК-7.1, ОПК-7.2
2.3 Свет, материалы и камера		2	4/2И			Самостоятельная проработка темы	Опрос по принципам работы	ОПК-7.1, ОПК-7.2
Итого по разделу		6	14/2,2 И		10			
3. Интерфейс и завершение проекта								
3.1 Пользовательский интерфейс (UI)	3	2	4		10	Самостоятельная проработка темы	Опрос по принципам работы	ОПК-7.1, ОПК-7.2
3.2 Анимация и эффекты		2	4		13	Самостоятельная проработка темы	Опрос по принципам работы	ОПК-7.1, ОПК-7.2
3.3 Сцены, звук и сборка проекта		2	2/2И			Самостоятельная проработка темы	Опрос по принципам работы	ОПК-7.1, ОПК-7.2
Итого по разделу		6	10/2И		23			
Итого за семестр		18	36/10,2 И		53		зачёт	
Итого по дисциплине		18	36/10,2 И		53		зачет	

## **5 Образовательные технологии**

Проведение лекционных занятий проводится в форме:

1. Информационных лекций.
2. Лекций-дискуссий.
3. Лекций с приглашенным экспертом.

На всех лекциях изложение содержания сопровождается презентацией, содержащих текстовые, иллюстративные, графические и видеоматериалы.

Лабораторные работы выполняются в двух уровнях сложности: сначала для ознакомления с технологией в форме кейсов, затем - в форме проектов.

При обучении используются информационно-коммуникационные образовательные технологии, под которыми понимается организация

образовательного

процесса, основанная на применении специализированных программных сред и технических средств работы с информацией.

Во время самостоятельной работы проводятся офлайн семинары с взаимной оценкой работ обучающихся.

На лабораторных работах и во время самостоятельной работы обучающиеся работают с ресурсами и сервисами образовательного портала

<https://newlms.magtu.ru>

## **6 Учебно-методическое обеспечение самостоятельной работы обучающихся**

Представлено в приложении 1.

## **7 Оценочные средства для проведения промежуточной аттестации**

Представлены в приложении 2.

## **8 Учебно-методическое и информационное обеспечение дисциплины**

### **а) Основная литература:**

1. Назарова Е. К. Погружение в индустрию компьютерных игр : практикум [для вузов] / Е. К. Назарова, Л. В. Курзаева ; Магнитогорский гос. технический ун-т им. Г. И. Носова. - Магнитогорск : МГТУ им. Г. И. Носова, 2022. - 1 CD-ROM. - Загл. с титул. экрана. - URL: <https://host.megaprolib.net/MP0109/Download/MObject/3317>. - ISBN

### **б) Дополнительная литература:**

1. Подбельский В.В. Программирование. Базовый курс C# : учебник для вузов / Вадим Валериевич Подбельский ; В. В. Подбельский. - Москва : Юрайт, 2025. - 369 с. - (Высшее образование). - URL: <https://urait.ru/bcode/560848> (дата обращения: 10.09.2025). - URL: <https://urait.ru/bcode/560848>. - URL: <https://urait.ru/book/cover/7A11FF4F-AB6A-42FE-AF6E-80965F641F1C>. - ISBN 978-5-534-10616-9. - дата обращения: 27.02.2026

Программирование. Базовый курс C# : учебник для вузов / Вадим Валериевич Подбельский ; В. В. Подбельский. - Москва : Юрайт, 2025. - 369 с. - (Высшее образование). - URL: <https://urait.ru/bcode/560848> (дата обращения: 10.09.2025). - URL: <https://urait.ru/bcode/560848>. - URL: <https://urait.ru/book/cover/7A11FF4F-AB6A-42FE-AF6E-80965F641F1C>. - ISBN 978-5-534-10616-9. - дата обращения: 27.02.2026

2. Преображенская Е. В. Создание трёхмерного проекта в Unity : учебно-методическое пособие / Е. В. Преображенская, А. А. Лим ; Преображенская Е. В., Лим А. А. - Москва : РТУ МИРЭА, 2024. - 76 с. - Книга из коллекции РТУ МИРЭА - Инженерно-технические науки. - URL: <https://e.lanbook.com/book/432683>. - URL: <https://e.lanbook.com/img/cover/book/432683.jpg>. - ISBN 978-5-7339-2261-4. - дата обращения: 27.02.2026

**в) Методические указания:** в приложении.

**г) Программное обеспечение и Интернет-ресурсы:**

**Программное обеспечение**

Наименование ПО	№ договора	Срок действия лицензии
MS Visual Studio 2017 Community Edition	свободно распространяемое ПО	бессрочно

**Профессиональные базы данных и информационные справочные системы**

Название курса	Ссылка
Национальная информационно-аналитическая система – Российский индекс научного цитирования (РИНЦ)	URL: <a href="https://elibrary.ru/project_risc.asp">https://elibrary.ru/project_risc.asp</a>
Электронные ресурсы библиотеки МГТУ им. Г.И. Носова	<a href="https://host.megaprolib.net/M/P0109/Web">https://host.megaprolib.net/M/P0109/Web</a>

**9 Материально-техническое обеспечение дисциплины (модуля)**

Материально-техническое обеспечение дисциплины включает:

Учебные аудитории для проведения занятий лекционного типа: специализированная (учебная) мебель (столы, стулья, доска аудиторная), мультимедийное оборудование (проектор, компьютер, экран) для презентации учебного материала по дисциплине;

Учебные аудитории для проведения лабораторных занятий, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации: специализированная (учебная) мебель (столы, стулья, доска аудиторная), персональные компьютеры объединенные в локальные сети с

выходом в

Internet и с доступом в электронную информационно-образовательную среду университета, оснащенные современными программно-методическими

комплексами

Аудитории для самостоятельной работы (компьютерные классы; читальные залы библиотеки): специализированная (учебная) мебель (столы, стулья, доска аудиторная), персональные компьютеры объединенные в локальные сети с

выходом в

Internet и с доступом в электронную информационно-образовательную среду университета, оснащенные современными программно-методическими

комплексами

Помещение для хранения и профилактического обслуживания учебного оборудования: мебель (столы, стулья, стеллажи для хранения учебно-наглядных пособий и учебно-методической документации), персональные компьютеры

### 6. Учебно-методическое обеспечение самостоятельной работы обучающихся

В рамках контрольной работы предполагается глубокая проработка тем.

#### Раздел 1. Основы работы в игровом движке

##### 1.1 Введение в Unity и интерфейс (2 часа)

- Задание: Создание первой сцены «Diorama».
- Практические шаги:
  1. Настройка проекта (3D Core).
  2. Создание ландшафта из примитивов (Cube, Sphere, Plane).
  3. Освоение инструментов трансформации (W, E, R).
  4. Иерархия объектов: создание вложенности (Parent-Child).
- Результат: Собранная статичная сцена (например, домик с деревом).

##### 1.2 Основы программирования на C# (2 часа)

- Задание: Скрипт-контроллер для вывода логов и изменения свойств.
- Практические шаги:
  1. Создание переменных разных типов (float, string, bool).
  2. Использование атрибута [SerializeField] для отображения в инспекторе.
  3. Работа в методах Start() и Update().
  4. Логика: изменение цвета объекта через код при старте.
- Результат: Скрипт, меняющий параметры объекта без перезапуска сцены.

##### 1.3 Физика и взаимодействие (2 часа)

- Задание: Полоса препятствий с использованием Rigidbody.
- Практические шаги:
  1. Настройка компонентов Rigidbody и Collider.
  2. Использование физических материалов (Bouncy, Friction).
  3. Написание скрипта с методом OnCollisionEnter (уничтожение объекта при столкновении).
- Результат: Сцена, где шарик падает, отскакивает и исчезает при касании «опасной» зоны.

#### Раздел 2. Механики и визуализация

##### 2.1 Ввод (Input System) и движение (2 часа)

- Задание: Создание управляемого персонажа.
- Практические шаги:
  1. Настройка осей в Input Manager (Horizontal/Vertical).
  2. Движение через transform.Translate или Rigidbody.AddForce.
  3. Ограничение зоны движения (условия if для координат).
- Результат: Персонаж, перемещающийся по нажатию WASD/стрелок.

##### 2.2 Префабы и управление ресурсами (2 часа)

- Задание: Система «Спаawner объектов».
- Практические шаги:
  1. Создание Prefab из настроенного объекта.
  2. Динамическое создание объектов через Instantiate по таймеру или нажатию клавиши.
  3. Организация структуры папок в окне Project.
- Результат: Бесконечная генерация монет или препятствий на сцене.

##### 2.3 Свет, материалы и камера (2 часа)

- Задание: Настройка атмосферы уровня.
- Практические шаги:
  1. Работа с Skybox и типами источников света (Directional, Point, Spot).
  2. Создание материалов (Albedo, Metallic, Smoothness, Emission).
  3. Настройка скрипта «Follow Camera» для слежения за игроком.

- Результат: Визуально проработанная сцена с настроенным освещением и камерой.

### Раздел 3. Интерфейс и завершение проекта

#### 3.1 Пользовательский интерфейс (UI) (2 часа)

- Задание: Создание игрового HUD и главного меню.
- Практические шаги:
  1. Создание Canvas и настройка Scaler.
  2. Добавление текста (TextMeshPro) и полосы здоровья (Image Fill Amount).
  3. Реализация кнопки «Начать заново» через UnityEngine.SceneManagement.
- Результат: Экран с отображением очков и рабочей кнопкой перезапуска.

#### 3.2 Анимация и эффекты (2 часа)

- Задание: Оживление объектов и системы частиц.
- Практические шаги:
  1. Создание анимации через окно Animation (изменение позиции/размера).
  2. Настройка переходов в Animator.
  3. Добавление Particle System для визуализации сбора предметов (вспышка).
- Результат: Плавные движения врагов и визуальный отклик на действия

игрока.

#### 3.3 Сцены, звук и сборка (2 часа)

- Задание: Финализация игры и экспорт.
- Практические шаги:
  1. Добавление компонентов AudioSource и AudioListener.
  2. Настройка фоновой музыки и звуков шагов.
  3. Build Settings: настройка иконки игры и сборка в формат .exe (Windows).
- Результат: Готовый файл игры, работающий вне редактора Unity.

Для самоконтроля можно воспользоваться следующими вопросами:

#### Раздел 1. Основы работы в игровом движке

##### 1.1 Введение в Unity и интерфейс

В чем разница между окнами Scene и Game?

Какую роль выполняет окно Hierarchy и как оно связано с окном Project?

Что такое Inspector и как добавить в него новый компонент?

Какие три базовые трансформации (Transform) есть у любого объекта?

Как изменить систему координат в редакторе (Local vs Global) и на что это влияет?

Зачем нужно сохранять сцены как отдельные файлы .unity?

Что произойдет с изменениями в инспекторе, если вносить их в режиме Play Mode?

Как сфокусировать камеру редактора на выбранном объекте (горячая клавиша)?

Как создать родительскую связь между объектами и что наследует «ребенок»?

Какие примитивы (Primitive Objects) доступны в Unity по умолчанию?

##### 1.2 Основы программирования на C#

В чем разница между методами Start() и Awake()?

Почему метод Update() вызывается чаще, чем FixedUpdate()?

Для чего используется атрибут [SerializeField] перед приватной переменной?

Что такое Component в контексте C# и как получить доступ к другому скрипту через GetComponent?

Какая разница между типами данных int, float и bool?

Зачем в коде использовать Time.deltaTime при расчете движения?

Как вывести отладочное сообщение в консоль Unity?

Что такое «нулевая ссылка» (NullReferenceException) и почему она возникает?

Какие основные математические операции доступны в классе Mathf?

Как объявить публичную переменную, чтобы она появилась в инспекторе?

##### 1.3 Физика и взаимодействие объектов

За что отвечает компонент Rigidbody?

В чем разница между Collision (столкновение) и Trigger (триггер)?  
Как сделать объект «кинематическим» (Is Kinematic) и на что это влияет?  
Какие основные типы коллайдеров существуют (Box, Sphere, Capsule, Mesh)?  
Зачем нужны Physics Materials (физические материалы)?  
Как приложить силу к объекту через код (метод AddForce)?  
Почему использование Mesh Collider на сложных объектах снижает производительность?  
Какие методы вызываются при входе, нахождении и выходе из триггера?  
Что такое слои (Layers) и как они помогают оптимизировать физику?  
Как предотвратить «проваливание» быстрых объектов сквозь стены (Collision Detection)?

## Раздел 2. Механики и визуализация

### 2.1 Ввод (Input System) и движение

Что такое оси (Axes) в стандартном Input Manager и какие значения они возвращают?  
В чем разница между GetKeyDown, GetKey и GetKeyUp?  
Как реализовать движение персонажа через transform.Translate?  
Почему для физического перемещения лучше использовать Rigidbody.MovePosition?  
Как ограничить вращение игрока по определенным осям через Constraints?  
Что такое вектор (Vector3) и как получить направление «вперед» относительно объекта?  
Как нормализовать вектор движения и зачем это нужно?  
Как реализовать прыжок персонажа с проверкой «на земле» (Grounded)?  
В чем отличие старой системы ввода от новой Input System Package?  
Как заставить объект плавно поворачиваться в сторону движения (Quaternion.LookRotation)?

### 2.2 Префабы и управление ресурсами

Что такое Prefab и в каких случаях его необходимо использовать?  
Как применить изменения, сделанные в экземпляре префаба на сцене, ко всем остальным копиям (Overrides)?  
Что такое Nested Prefabs (вложенные префабы)?  
Как динамически создать объект из кода (метод Instantiate)?  
Как правильно удалить объект со сцены во время игры (метод Destroy)?  
Что такое Resources.Load и почему его рекомендуют использовать с осторожностью?  
Зачем нужны теги (Tags) и как найти все объекты с определенным тегом?  
В чем разница между Active и Inactive состоянием объекта в иерархии?  
Как создать префаб из обычного игрового объекта?  
Что такое Variant Prefab и чем он отличается от оригинала?

### 2.3 Свет, материалы и камера

Какие 4 основных типа источников света есть в Unity?  
В чем разница между Realtime и Baked освещением?  
За что отвечает параметр Albedo в стандартном материале?  
Как создать эффект самосвечения объекта через карту Emission?  
Что такое Skybox и как его сменить на сцене?  
Какую роль играет компонент Audio Listener на камере?  
Как изменить поле зрения (Field of View) камеры и на что это влияет?  
Что такое Occlusion Culling и как оно помогает в оптимизации?  
Как программно заставить камеру плавно следовать за игроком (Lerp или SmoothDamp)?  
Зачем нужен параметр Clear Flags в настройках камеры?

## Раздел 3. Интерфейс и завершение проекта

### 3.1 Пользовательский интерфейс (UI)

Что такое Canvas и какие режимы отрисовки (Render Mode) у него есть?  
 Как работает компонент RectTransform в отличие от обычного Transform?  
 Зачем нужны якоря (Anchors) и точки опоры (Pivot) в интерфейсе?  
 Как реализовать отклик кнопки (UI Button) на нажатие через событие OnClick?  
 В чем преимущество TextMeshPro перед стандартным компонентом Text?  
 Как сделать прогресс-бар (например, HP) с помощью компонента Image?  
 Как включить или выключить UI-панель из скрипта?  
 Что такое Event System и почему без него интерфейс не будет кликабельным?  
 Как адаптировать интерфейс под разные разрешения экрана (UI Scaler)?  
 Как отобразить 3D-объект «поверх» всех окон интерфейса?

### 3.2 Анимация и эффекты

Что такое Animator Controller и как он связан с клипами анимации?  
 Как создать переход между анимациями (Transition) и что такое Parameters?  
 В чем разница между состояниями анимации в режиме Loop и без него?  
 Как запустить анимацию из скрипта (например, по триггеру)?  
 Что такое Animation Events и как с их помощью вызвать функцию в коде?  
 Из каких основных модулей состоит Particle System (Эмиссия, Форма, Цвет)?  
 Как сделать так, чтобы частицы сталкивались с полом (модуль Collision)?  
 Как реализовать «вспышку» частиц при смерти врага?  
 Что такое Sprite Editor и как подготовить 2D-анимацию из листа (Spritesheet)?  
 Как использовать Blend Trees для плавного смешивания анимаций ходьбы и бега?

### 3.3 Сцены, звук и сборка проекта

Как добавить новую сцену в финальную сборку (Build Settings)?  
 Какая функция используется для загрузки новой сцены по имени или индексу?  
 В чем разница между фоновым звуком (2D) и пространственным звуком (3D)?  
 Как реализовать плавное затухание звука при отдалении от источника?  
 Что такое Build Target и на какие платформы можно собрать проект?  
 Как создать иконку для своего приложения в настройках Player Settings?  
 Зачем нужно сохранять данные между сценами и как работает DontDestroyOnLoad?  
 Как проверить производительность игры в реальном времени через окно Profiler?  
 Что такое PlayerPrefs и как с их помощью сохранить рекорд игрока?  
 Какие файлы и папки нужно передавать другому человеку, чтобы он мог открыть ваш проект?

### **Методические рекомендации для самостоятельной работы студентов**

В процессе выполнения самостоятельной работы студенты должны научиться воспринимать сведения на слух, фиксировать информацию в виде записей в тетрадах, работать с письменными текстами, самостоятельно извлекая из них полезные сведения и оформляя их в виде тезисов, конспектов, систематизировать информацию в виде заполнения таблиц, составления схем. Важно научиться выделять главные мысли в письменном тексте; определять свою позицию к полученным на занятиях сведениям, четко формулировать ее; аргументировать свою точку зрения: высказывать оценочные суждения; осуществлять самоанализ. Необходимо учиться владеть устной и письменной речью; вести диалог; участвовать в дискуссии; раскрывать содержание изучаемой проблемы в монологической речи; выступать с сообщениями и докладами.

**Подготовка к семинарским занятиям.** Семинар – один из основных видов практических занятий по гуманитарным дисциплинам. Он предназначен для углубленного изучения отдельных тем и курсов. По форме проведения семинары обычно представляют собой решение задач, обсуждение докладов, беседу по плану или дискуссию по проблеме.

Подготовка к занятиям заключается, прежде всего, в освоении того теоретического материала, который выносится на обсуждение. Для этого необходимо в первую очередь перечитать конспект лекции или разделы учебника, в которых присутствует установочная информация. Изучение рекомендованной литературы необходимо сделать максимально творчески – не просто укладывая в память новые сведения, а осмысливая и анализируя

материал. Закрепить свои знания можно с помощью записей, выписок или тезисного конспекта.

Если семинар представлен докладами, то основная ответственность за его проведение лежит на докладчиках. Как сделать это успешно смотрите в разделе «Доклад». Однако роль остальных участников семинара не должна быть пассивной. Студенты, прослушав доклад, записывают кратко главное его содержание и задают выступающему уточняющие вопросы. Чем более основательной была домашняя подготовка по теме, тем активнее происходит обсуждение проблемных вопросов. На семинаре всячески поощряется творческая, самостоятельная мысль, дается возможность высказать критические замечания.

Беседа по плану представляет собой заранее подготовленное совместное обсуждение вопросов темы каждым из участников. Эта форма потребует от студентов не только хорошей самостоятельной проработки теоретического материала, но и умение участвовать в коллективной дискуссии: кратко, четко и ясно формулировать и излагать свою точку зрения перед сокурсниками, отстаивать позицию в научном споре, присоединиться к чужому мнению или оппонировать другим участникам.

**Реферат** – самый простой и наименее самостоятельный вид письменной работы. Суть его состоит в кратком изложении содержащихся в научной литературе взглядов и идей по заданной теме. Реферат не требует оригинальности и новизны. В нем оценивается умение студента работать с книгой: выделять и формулировать проблему, отбирать основные тезисы и вспомогательные данные, логически выстраивать материал, грамотно оформлять научный текст.

Студентам предлагается два вида рефератных работ.

**Реферирование научной литературы** представляет собой сокращенное изложение содержания статьи или книги с основными сведениями и выводами. Такие рефераты актуальны тогда, когда в юридических источниках появляется новый теоретический или практический материал по изучаемой теме. От студента требуется, внимательно ознакомившись с первоисточником, максимально точно и полно передать его содержание. Для этого целесообразно выбрать форму последовательного изложения прочитанной книги, не меняя ее общий план и структуру (главы, разделы, параграфы). Необходимо сохранить логику повествования и позаботиться о связности текста. Авторские, оригинальные и новаторские мысли и идеи лучше передавать не своими словами, а с помощью цитирования. Объем реферата будет определяться содержанием источника, а также его научной и практической ценностью. Но в любом случае предпочтение отдается краткости и лаконичности, умению отбирать главное и освободиться от второстепенного.

Реферат по теме представляет обзор научных взглядов и концепций по проблемному вопросу в изучаемой теме.

Если вам предложена тема такого реферата на выбор, то предпочтение следует отдать той, которая для вас интересна или знакома. Она не должна быть очень сложной и объемной, в противном случае реферат будет напоминать курсовую работу.

Для подготовки реферата студенту необходимо самому или с участием преподавателя подобрать источники информации. Следует позаботиться, чтобы в вашем списке оказались не случайные, а ценные в информационном плане книги. Можно выполнить работу, обратившись и к одному источнику – пособию, монографии, исследованию. Но лучше, если вы обратитесь к двум-трем научным трудам – это позволит представить проблему с нескольких точек зрения и высказать личные предпочтения.

Одним из главных критериев оценки реферата будет соответствие его содержания заявленной теме. Для этого бегло ознакомившись с первоисточниками составьте предварительный план будущего реферата, обозначив в нем принципиально важные моменты и этапы освещения проблемы. После того, как у вас появятся рабочие записи по результатам изучения научной литературы и обширная информация по теме в целом, можно будет скорректировать общий план реферата. Старайтесь при работе над ним тщательно избавляться от «излишеств»: всякого рода абстрактных рассуждений,

чрезмерных подробностей и многочисленных примеров, которые «размывают» тему или уводят от неё.

Структура реферата включает в себя введение, основную часть и заключение. Во введении формулируются цели и задачи работы, ее актуальность. Основная часть представляет собой последовательное и аргументированное изложение различных точек зрения на проблему, ее анализ, предполагаемые пути решения. Заключение обобщает основные мысли или обосновывает перспективы дальнейшего исследования темы. Если реферат достаточно объемный, то потребуется разделение текста на разделы (главы, параграфы). Иллюстративный материал – таблицы, схемы, графики – могут располагаться как внутри основной части, так и в разделе «Приложение».

Объем реферата зависит от целей и задач, решаемых в работе – от 5 до 20 страниц машинописного текста через два интервала. Если в задании, выданном преподавателем объем не оговаривается, то следует исходить из разумной целесообразности.

В реферате в обязательном порядке размещаются титульный лист, план или оглавление работы, а также список используемой литературы.

Обычно реферат может зачитывается как письменная работа, но некоторые преподаватели практикуют публичную защиту рефератов или их «озвучивание» на семинарских занятиях. В этом случае необходимо приложить дополнительные усилия для подготовки публичного выступления по материалам рефератной работы.

**Доклад** представляет собой устную форму сообщения информации. Он используется в вузе на семинарских занятиях и на научных студенческих конференциях.

Подготовка доклада осуществляется в два этапа: написание письменного текста на заданную тему и подготовка устного выступления перед аудиторией слушателей с освещением этой темы. Письменный доклад оформляется как реферат.

При работе над докладом следует учесть некоторые специфические особенности:

- Объем доклада должен согласовываться со временем, отведенным для выступления.

- При выборе темы нужно учитывать не только собственные интересы, но и интересы потенциальных слушателей. Ваше сообщение необходимо согласовывать с уровнем знаний и потребностей публики.

- Подготовленный текст доклада должен хорошо восприниматься на слух. Даже если отобранный вами материал сложен и неоднозначен, говорить желательно просто и ясно, не перегружая речь наукообразными оборотами и специфическими терминами.

Следует отметить, что иногда преподаватель не требует от студентов письменного варианта доклада и оценивает их работу исключительно по устному выступлению. Но значительно чаще письменный доклад проверяется и его качество также оценивается в баллах. Вне зависимости от того, нужно или не нужно будет сдавать на проверку текст будущего выступления, советуем не отказываться от письменной записи доклада. Это поможет избежать многих ошибок, которые случаются во время устной импровизации: отклонение от темы, нарушения логической последовательности, небрежное обращение с цитатами, злоупотребление деталями и т.д. Если вы хорошо владеете навыками свободной речи и обладаете высокой культурой мышления, то замените письменный доклад составлением тезисного плана. С его помощью зафиксируйте основные мысли и идеи, выстройте логику повествования, отберите яркие и точные примеры, сформулируйте выводы.

При подготовке к устному выступлению возьмите на вооружение некоторые советы:

- Лучший вариант выступления перед аудиторией – это свободная речь, не осложненная чтением текста. Но если у вас не выработано умение общаться с публикой без бумажки, то не пытайтесь сделать это сразу, без подготовки. Осваивать этот опыт нужно постепенно, от доклада к докладу увеличивая объем речи без заглядывания в текст.

- Если вы намерены считать доклад с подготовленных письменных записей, то постарайтесь, чтобы чтение было «художественным»: обозначайте паузой логические переходы от части к части, выделяйте интонационно особо важные мысли и аргументы,

варьируйте темп речи.

– Читая доклад, не торопитесь, делайте это как можно спокойнее. Помните, что скорость произношения текста перед слушателями всегда должна быть более медленной, чем скорость вашей повседневной речи.

– Сверьте письменный текст с хронометром, для этого прочитайте его несколько раз с секундомером в руках. В случае, если доклад окажется слишком длинным или коротким, проведите его реконструкцию. Однако вместе с сокращениями или дополнениями не «потеряйте» тему. Не поддавайтесь искушению рассказать все, что знаете – полно и подробно.

– Обратите внимание на тембр и силу вашего голоса. Очень важно, чтобы вас было слышно в самых отдаленных частях аудитории, и при этом вы не «глушили» вблизи вас находящихся слушателей. Варьируйте тембр речи, он придаст ей выразительность и поможет избежать монотонности.

– Следите за своими жестами. Чрезмерная жестикуляция отвлекает от содержания доклада, а полное ее отсутствие снижает действенную силу выступления. Постарайтесь избавиться от жестов, демонстрирующих ваше волнение (когда крутятся ручки, теребятся пуговицы, заламываются пальцы). Используйте жесты – выразительные, описательные, подражательные, указующие – для полноты передачи ваших мыслей.

– Установите зрительный контакт с аудиторией. Не стоит все время смотреть в окно, опускать глаза или сосредотачиваться на тексте. Старайтесь зрительно общаться со всеми слушателями, переводя взгляд от одних к другим. Не обращайтесь к замечаниям опоздавших и не прерывайте свой доклад замечаниями. Но вместе с тем следите за реакцией публики на ваше выступление (одобрение, усталость, интерес, скуку) и если сможете, вносите коррективы в речь с целью повышения интереса к его содержанию.

– Отвечать на вопросы в конце выступления надо кратко, четко и уверенно, без лишних подробностей и повторов. Постарайтесь предугадать возможные вопросы своих слушателей и подготовиться к ним заранее. Но если случится, что вы не знаете ответа на заданный вам вопрос, не бойтесь в этом признаться. Это значительно лучше, чем отвечать не по существу или отшучиваться.

– Проведите генеральную репетицию своего доклада перед друзьями или близкими. Это поможет заранее выявить некоторые недостатки – стилистически слабые места, труднопроизносимые слова и фразы, затянутые во времени части и т.д. Проанализируйте свою дикцию, интонации, жесты. Сделайте так, чтобы они помогали, а не мешали успешно представить публике подготовленный вами доклад.

**Презентация** – современный способ устного или письменного представления информации с использованием мультимедийных технологий.

Существует несколько вариантов презентаций:

- Презентация с выступлением докладчика.
- Презентация с комментариями докладчика.
- Презентация для самостоятельного просмотра, которая может демонстрироваться перед аудиторией без участия докладчика.

**Подготовка презентации включает в себя несколько этапов:**

### **1. Планирование презентации**

От ответов на эти вопросы будет зависеть всё построение презентации:

- каково предназначение и смысл презентации (демонстрация результатов научной работы, защита дипломного проекта и т.д.);
- какую роль будет выполнять презентация в ходе выступления (сопровождение доклада или его иллюстрация);
- какова цель презентации (информирование, убеждение или анализ);
- на какое время рассчитана презентация (короткое – 5-10 минут или продолжительное – 15-20 минут);
- каков размер и состав зрительской аудитории (10-15 человек или 80-100; преподаватели, студенты или смешанная аудитория).

## **2. Структурирование информации**

– в презентации не должна быть менее 10 слайдов, а общее их количество превышать 20 - 25.

– основными принципами при составлении презентации должны быть ясность, наглядность, логичность и запоминаемость;

– презентация должна иметь сценарий и четкую структуру, в которой будут отражены все причинно-следственные связи,

– работа над презентацией начинается после тщательного обдумывания и написания текста доклада, который необходимо разбить на фрагменты и обозначить связанные с каждым из них задачи и действия;

– первый шаг – это определение главной идеи, вокруг которой будет строиться презентация;

– часть информации можно перевести в два типа наглядных пособий: текстовые, которые помогут слушателям следить за ходом развертывания аргументов и графические, которые иллюстрируют главные пункты выступления и создают эмоциональные образы.

– сюжеты презентации могут разъяснять или иллюстрировать основные положения доклада в самых разнообразных вариантах.

Очень важно найти правильный баланс между речью докладчика и сопровождающими её мультимедийными элементами.

Для этого целесообразно:

– определить, что будет представлено на каждом слайде, что будет в это время говориться, как будет сделан переход к следующему слайду;

– самые важные идеи и мысли отразить и на слайдах и произнести словами, тогда как второстепенные – либо словами, либо на слайдах;

– информацию на слайдах представить в виде тезисов – они сопровождают подробное изложение мыслей выступающего, а не наоборот;

– для разъяснения положений доклада использовать разные виды слайдов: с текстом, с таблицами, с диаграммами;

– любая презентация должна иметь собственную драматургию, в которой есть:

1. «завязка» – пробуждение интереса аудитории к теме сообщения (яркий наглядный пример);

2. «развитие» – демонстрация основной информации в логической последовательности (чередование текстовых и графических слайдов);

3. «кульминация» – представление самого главного, нового, неожиданного (эмоциональный речевой или иллюстративный образ);

4. «развязка» – формулирование выводов или практических рекомендаций (видеоряд).

## **3. Оформление презентации**

Оформление презентации включает в себя следующую обязательную информацию:

Титульный лист

– - представляет тему доклада и имя автора (или авторов);

– на защите курсовой или дипломной работы указывает фамилию и инициалы научного руководителя или организации;

– на конференциях обозначает дату и название конференции.

План выступления:

– формулирует основное содержание доклада (3-4 пункта);

– фиксирует порядок изложения информации;

Содержание презентации:

– включает текстовую и графическую информацию;

– иллюстрирует основные пункты сообщения;

– может представлять самостоятельный вариант доклада;

Завершение:

– обобщает, подводит итоги, суммирует информацию;

– может включать список литературы к докладу;

- содержит слова благодарности аудитории.

#### **4. Дизайн презентации**

##### **Текстовое оформление**

- Не стоит заполнять слайд слишком большим объемом информации - лучше всего запоминаются не более 3-х фактов, выводов, определений.
- Оптимальное число строк на слайде – 6 -11.
- Короткие фразы запоминаются визуальнo лучше. Пункты перечней не должны превышать двух строк на фразу.
- Наибольшая эффективность достигается тогда, когда ключевые пункты отображаются по одному на каждом отдельном слайде
- Если текст состоит из нескольких абзацев, то необходимо установить крас-ную строку и интервал между абзацами.
- Ключевые слова в информационном блоке выделяются цветом, шрифтом или композиционно.
- Информацию предпочтительнее располагать горизонтально, наиболее важную - в центре экрана.
- Не следует злоупотреблять большим количеством предлогов, наречий, прилагательных, вводных слов.
- Цифровые материалы лучше представить в виде графиков и диаграмм – таблицы с цифровыми данными на слайде воспринимаются плохо.
- Необходимо обратить внимание на грамотность написания текста. Ошибки во весь экран производят неприятное впечатление

##### **Шрифтовое оформление**

- Шрифты без засечек (Arial, Tahoma, Verdana) читаются легче, чем гротески. Нельзя смешивать различные типы шрифтов в одной презентации.
- Шрифтовой контраст можно создать посредством размера шрифта, его толщины, начертания, формы, направления и цвета;
- Для заголовка годится размер шрифта 24-54 пункта, а для текста - 18-36 пунктов.
- Курсив, подчеркивание, жирный шрифт используются ограниченно, только для смыслового выделения фрагментов текста.
- Для основного текста не рекомендуются прописные буквы.

##### **Цветовое оформление**

- На одном слайде не используется более трех цветов: фон, заголовок, текст.
- Цвет шрифта и цвет фона должны контрастировать – текст должен хорошо читаться, но не резать глаза.
- Для фона предпочтительнее холодные тона.
- Существуют не сочетаемые комбинации цветов. Об этом можно узнать в специальной литературе.
- Черный цвет имеет негативный (мрачный) подтекст. Белый на черном читается плохо.
- Если презентация большая, то есть смысл разделить её на части с помощью цвета – разный цвет способен создавать разный эмоциональный настрой.
- Нельзя выбирать фон, который содержит активный рисунок.

##### **Композиционное оформление**

- Следует соблюдать единый стиль оформления. Он может включать определенный шрифт (гарнитура и цвет), фон цвета или фоновый рисунок, декоративный элемент небольшого размера и т.д.
- Не приемлемы стили, которые будут отвлекать от презентации.
- Крупные объекты в композиции смотрятся неважно.
- Вспомогательная информация (управляющие кнопки) не должна преобладать

над основной (текстом и иллюстрацией).

- Для серьезной презентации отбираются шаблоны, выполненные в деловом стиле.

#### **Анимационное оформление**

- Основная роль анимации – дозирования информации. Аудитория, как правило, лучше воспринимает информацию порциями, небольшими зрительными фрагментами.

- Анимация используется для привлечения внимания или демонстрации развития какого-либо процесса

- Не стоит злоупотреблять анимационными эффектами, которые отвлекают от содержания или утомляют глаза читающего.

- Особенно нежелательно частое использование таких анимационных эффектов как вылет, вращение, волна, побуквенное появление текста.

#### **Звуковое оформление**

- Музыкальное сопровождение призвано отразить суть или подчеркнуть особенности темы слайда или всей презентации, создать определенный эмоциональный настрой.

- Музыка целесообразно включать тогда, когда презентация идет без словесного сопровождения.

- Звуковое сопровождение используется только по необходимости, поскольку даже фоновая тихая музыка создает излишний шум и мешает восприятию содержания.

- Необходимо выбрать оптимальную громкость, чтобы звук был слышан всем слушателем, но не был оглушительным.

#### **Графическое оформление**

- Рисунки, фотографии, диаграммы призваны дополнить текстовую информацию или передать её в более наглядном виде.

- Нельзя представлять рисунки и фото плохого качества или с искаженными пропорциями.

- Желательно, чтобы изображение было не столько фоном, сколько иллюстрацией, равной по смыслу самому тексту, чтобы помочь по-новому понять и раскрыть его.

- Следует избегать некорректных иллюстраций, которые неправильно или двусмысленно отражают смысл информации.

- Необходимо позаботиться о равномерном и рациональном использовании пространства на слайде: если текст первичен, то текстовый фрагмент размещается в левом верхнем углу, а графический рисунок внизу справа и наоборот.

- Иллюстрации рекомендуется сопровождать пояснительным текстом. Подписи к картинкам лучше выполнять сбоку или снизу, если это только не название самого слайда.

- Если графическое изображение используется в качестве фона, то текст на этом фоне должен быть хорошо читаем.

#### **Таблицы и схемы**

- Не стоит вставлять в презентацию большие таблицы – они трудны для восприятия. Лучше заменить их графиками, построенными на основе этих таблиц.

- Если все же таблицу показать надо, то следует оставить как можно меньше строк и столбцов, отобрав и разместив только самые важные данные.

- При использовании схем на слайдах необходимо выравнивать ряды блоков схемы, расстояние между блоками, добавить соединительные линии при помощи инструментов Автофигур.

- При создании схем нужно учитывать связь между составными частями схемы: если они равнозначны, то заполняются одним шрифтом, фоном и текстом, если есть первостепенная информация, то она выделяется особым способом с помощью

организационных диаграмм.

#### **Аудио и видео оформление**

- Видео, кино и теле материалы могут быть использованы полностью или фрагментарно в зависимости от целей, которые преследуются.
- Продолжительность фильма не должна превышать 15-25 минут, а фрагмента – 4-6 минут.
- Нельзя использовать два фильма на одном мероприятии, но показать фрагменты из двух фильмов вполне возможно.

**Подготовка к зачёту.** Готовиться к зачёту нужно заранее и в несколько этапов. Для этого:

- Просматривайте конспекты лекций сразу после занятий. Это поможет разобраться с непонятными моментами лекции и возникшими вопросами, пока еще лекция свежа в памяти.
- Бегло просматривайте конспекты до начала следующего занятия. Это позволит «освежить» предыдущую лекцию и подготовиться к восприятию нового материала.
- Каждую неделю отводите время для повторения пройденного материала.
- Непосредственно при подготовке:
- Упорядочьте свои конспекты, записи, задания.
- Прикиньте время, необходимое вам для повторения каждой части (блока) материала, выносимого на зачет.
- Составьте расписание с учетом скорости повторения материала, для чего
- Разделите вопросы для зачёта на знакомые (по лекционному курсу, семинарам, конспектированию), которые потребуют лишь повторения и новые, которые придется осваивать самостоятельно. Начните с тем хорошо вам известных и закрепите их с помощью конспекта и учебника. Затем пополните свой теоретический багаж новыми знаниями, обязательно воспользовавшись рекомендованной литературой.
- Правильно используйте консультации, которые проводит преподаватель. Приходите на них с заранее проработанными самостоятельно вопросами. Вы можете получить разъяснение по поводу сложных, не до конца понятых тем, но не рассчитывайте во время консультации на исчерпывающую информации по содержанию всего курса.

7. Оценочные средства для проведения промежуточной аттестации

а) Планируемые результаты обучения и оценочные средства для проведения промежуточной аттестации:

Код индикатора	Индикатор достижения компетенции	Оценочные средства
ОПК-7 Способен разрабатывать алгоритмы и программы, пригодные для практического применения;		
ОПК-7.1	Определяет средства разработки программных средств для решения практических задач профессиональной деятельности	<p><b>Теоретические вопросы к зачету:</b></p> <ol style="list-style-type: none"> <li>1. В каких ситуациях для реализации перемещения персонажа профессионально оправдано использование компонента CharacterController, а в каких — связку Rigidbody + Collider?</li> <li>2. Сравните возможности Input Manager и Input System Package: какое из этих средств вы выберете для проекта с поддержкой кроссплатформенного управления (геймпад + сенсор + клавиатура)?</li> <li>3. Какой графический конвейер (URP или HDRP) следует определить как основной при разработке мобильного приложения, и какими техническими ограничениями продиктован этот выбор?</li> <li>4. Для реализации сложной логики поведения ИИ (врага) какое программное средство эффективнее: классические условия if-else или архитектурный паттерн State Machine (Конечный автомат)?</li> <li>5. Какое средство разработки в Unity вы определите как наиболее подходящее для реализации системы сохранений игрового прогресса: PlayerPrefs, JSON-сериализация или работа с бинарными файлами?</li> <li>6. Для оптимизации производительности в сцене с тысячами одинаковых объектов (например, трава или камни), какие инструменты движка следует задействовать: Static Batching, Dynamic Batching или GPU Instancing?</li> <li>7. Если проект предполагает подгрузку контента из облака или работу с огромным объемом ассетов, не помещающихся в память, какое программное решение предпочтительнее: папка Resources или система Addressables?</li> <li>8. Какое средство межобъектного взаимодействия в коде (прямые ссылки на объекты, интерфейсы или UnityEvents/C# Events) обеспечивает наилучшую масштабируемость проекта и почему?</li> </ol>

9. В каких случаях при настройке сборки (Build Settings) профессиональный разработчик должен определить IL2CPP как Scripting Backend вместо стандартного Mono?
10. С помощью каких инструментов внутри системы UGUI (Anchors, Pivot, Layout Groups) решается задача адаптации интерфейса под различные разрешения и соотношения сторон экранов?

**Практические задания:**

1. Разработка контроллера игрока с выбором системы перемещения: Создайте игровую сцену с персонажем и полосой препятствий. Реализуйте два варианта управления: первый — через компонент CharacterController (аркадное движение без физики), второй — через Rigidbody с использованием AddForce (физическое взаимодействие). В коде настройте переключение между ними и добавьте ограничение скорости через Vector3.ClampMagnitude, чтобы продемонстрировать понимание профессиональных инструментов контроля динамики объекта.
2. Создание системы динамического спавна с использованием префабов: Спроектируйте алгоритм «Генератор препятствий», который через случайные промежутки времени создает объекты из заранее настроенного Prefab в определенных точках пространства. Используйте метод Instantiate для генерации и настройте автоматическое удаление объектов через Destroy(obj, delay) при выходе за границы видимости камеры. Это задание проверяет навык эффективного управления ресурсами памяти и работы с шаблонами объектов.
3. Адаптивная настройка визуальной среды и материалов: Создайте интерьерную сцену и настройте в ней три типа источников света: Directional Light для имитации солнца, Point Light для ламп и Spot Light для направленных лучей. Создайте и примените материалы с использованием разных карт (Albedo, Metallic, Emission), а затем настройте камеру, добавив скрипт плавного следования за игроком с использованием метода Vector3.Lerp или SmoothDamp. Задание направлено на освоение инструментов визуализации и постобработки.
4. Реализация интерактивного UI и системы состояний: Разработайте игровой интерфейс (Canvas), включающий полосу здоровья (Health Bar) и текстовый счетчик очков. Напишите скрипт-менеджер, который

		<p>обновляет данные интерфейса при срабатывании событий столкновения игрока с бонусами или врагами (используя OnTriggerEnter). Реализуйте логику «Game Over»: при достижении нулевого здоровья экран блокируется, и появляется кнопка перезагрузки сцены, работающая через SceneManager.</p> <p><b>Комплексное задание</b></p> <ol style="list-style-type: none"> <li>1. Разработка собственное игры</li> </ol>
ОПК-7.2	<p>Разрабатывает алгоритмы и программы для решения прикладных задач различных классов</p>	<p><b>Теоретические вопросы к зачету:</b></p> <ol style="list-style-type: none"> <li>1. Опишите алгоритм реализации прыжка персонажа: какие проверки (Raycast, Collision или Trigger) и физические расчеты необходимы, чтобы избежать «бесконечного прыжка» в воздухе?</li> <li>2. Как математически реализовать алгоритм плавного слежения камеры за игроком без использования сторонних плагинов (сравните подходы Lerp и SmoothDamp)?</li> <li>3. Каков алгоритм работы спавнера объектов, исключаящий их наложение друг на друга при генерации в случайном диапазоне координат?</li> <li>4. Объясните логику работы алгоритма Timer/Cooldown: как программно ограничить частоту выстрелов или использования способности игроком, используя Time.time или Time.deltaTime?</li> <li>5. Опишите алгоритм поиска пути и преследования цели для простого ИИ: какие проверки дистанции и векторы направления (target.position - transform.position) должны лежать в его основе?</li> <li>6. Как реализовать алгоритм инвентарной системы: какие структуры данных (массивы, списки List&lt;&gt; или словари Dictionary&lt;&gt;) наиболее эффективны для хранения и поиска предметов по ID?</li> <li>7. Объясните алгоритм расчета урона с учетом брони: как программно связать переменные здоровья, силы атаки и коэффициента защиты в единую формулу внутри метода TakeDamage?</li> <li>8. Опишите логику алгоритма смены сцен с сохранением данных: как передать информацию о количестве очков или текущем уровне здоровья из одной сцены в другую?</li> <li>9. Каков алгоритм реализации циклического движения платформы между двумя точками (A и B) с использованием тригонометрических функций (Mathf.PingPong или Mathf.Sin)?</li> <li>10. Опишите алгоритм обработки ввода: как разделить логику считывания клавиш (Input) и</li> </ol>

		<p>логику выполнения действий (Movement/Action), чтобы код оставался чистым и расширяемым?</p> <p><b>Практические задания:</b></p> <ol style="list-style-type: none"> <li>1. Разработка алгоритма патрулирования ИИ: Напишите программу для вражеского объекта, который перемещается по заданному массиву точек (Waypoints). Реализуйте логику: при достижении точки объект ждет 2 секунды, а затем выбирает следующую; если игрок входит в радиус обнаружения (проверка через Vector3.Distance), алгоритм переключается с патрулирования на преследование цели.</li> <li>2. Создание системы циклической генерации уровня: Разработайте алгоритм «бесконечного коридора» или платформера. Программа должна динамически создавать новые сегменты уровня перед игроком и удалять пройденные позади него, используя объектные пулы или метод Instantiate/Destroy. Алгоритм должен гарантировать бесшовную стыковку сегментов по координате Z или X.</li> <li>3. Программирование математической модели снаряда: Реализуйте алгоритм полета снаряда (пули или стрелы) без использования стандартной гравитации Unity. Программа должна рассчитывать траекторию через transform.Translate с учетом заданного вектора направления и постепенного снижения по оси Y (имитация баллистики), а также определять момент столкновения с целью через Raycast.</li> <li>4. Разработка алгоритма сбора и обработки игровых данных: Создайте систему достижений. Напишите программу-менеджер, которая подписывается на события сбора монет или уничтожения врагов. Алгоритм должен накапливать значения в переменных, проверять условия (например, «собрано 10 монет») и при выполнении условия выводить уведомление на экран и сохранять результат в PlayerPrefs.</li> <li>5.</li> </ol> <p><b>Комплексное задание</b> Разработка собственной игры</p>
--	--	---

б) Порядок проведения промежуточной аттестации, показатели и критерии оценивания:

Промежуточная аттестация по дисциплине проводится в форме зачета (защита проекта)

**Критерии оценки зачета**

– «зачтено» – обучающийся демонстрирует высокий уровень сформированности компетенций, свободно выполняет практические задания и применяет полученные умения в

ситуациях повышенной сложности.

– **«не зачтено»** – обучающийся не может показать знания на уровне воспроизведения и объяснения информации, не может показать интеллектуальные навыки решения простых задач.