

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение высшего образования  
"Казанский (Приволжский) федеральный университет"  
Институт информационных технологий и интеллектуальных систем



УТВЕРЖДАЮ

Проректор по образовательной деятельности КФУ

Е.А. Турилова

28 февраля 2025 г.

подписано электронно-цифровой подписью

## Программа дисциплины

### Разработка высокопроизводительных игровых систем

Направление подготовки: 09.03.04 - Программная инженерия

Профиль подготовки: Индустрія разработки видеоигр

Квалификация выпускника: бакалавр

Форма обучения: очное

Язык обучения: русский

Год начала обучения по образовательной программе: 2025

## **Содержание**

1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения ОПОП ВО
2. Место дисциплины (модуля) в структуре ОПОП ВО
3. Объем дисциплины (модуля) в зачетных единицах с указанием количества часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся
4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий
  - 4.1. Структура и тематический план контактной и самостоятельной работы по дисциплине (модулю)
  - 4.2. Содержание дисциплины (модуля)
5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)
6. Фонд оценочных средств по дисциплине (модулю)
7. Перечень литературы, необходимой для освоения дисциплины (модуля)
8. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)
9. Методические указания для обучающихся по освоению дисциплины (модуля)
10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем (при необходимости)
11. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)
12. Средства адаптации преподавания дисциплины (модуля) к потребностям обучающихся инвалидов и лиц с ограниченными возможностями здоровья
13. Приложение №1. Фонд оценочных средств
14. Приложение №2. Перечень литературы, необходимой для освоения дисциплины (модуля)
15. Приложение №3. Перечень информационных технологий, используемых для освоения дисциплины (модуля), включая перечень программного обеспечения и информационных справочных систем

Программу дисциплины разработал(а)(и): Надыршина К.Р. ; ассистент, б.с. Нурлыгаянов Н.Р. (Кафедра индустрии разработки видеоигр, Институт информационных технологий и интеллектуальных систем), NiRNurlygayanov@kpfu.ru ; ассистент, б.с. Шубин А.В. (Кафедра индустрии разработки видеоигр, Институт информационных технологий и интеллектуальных систем), AleVShubin@kpfu.ru

## **1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения ОПОП ВО**

Обучающийся, освоивший дисциплину (модуль), должен обладать следующими компетенциями:

Шифр компетенции	Расшифровка приобретаемой компетенции
ПК-8	Владение навыками использования различных технологий разработки программного обеспечения

Обучающийся, освоивший дисциплину (модуль):

Должен знать:

основы работы с Unreal Engine, включая структуру проекта, иерархию классов (Actor, Pawn, Character, PlayerController, AIController, GameMode) и баланс между Blueprint и C++.

принципы программирования на C++ в Unreal Engine, включая создание классов, работу с UPROPERTY, UFUNCTION, UClass, управление памятью и взаимодействие с игровыми объектами.

основы игровых механик, включая обработку ввода, физику объектов, работу с анимациями (AnimationBlueprint, BlendSpace) и AI (Behavior Trees, Blackboard).

Должен уметь:

разрабатывать игровые механики и взаимодействие объектов, используя Blueprint и C++, а также физические и анимационные системы Unreal Engine.

программировать AI и игровые события, используя AIController, Behavior Trees, AIMoveTo, LineTraceByChannel для управления NPC.

оптимизировать и отлаживать код, используя UE\_LOG, Debug Draw, трассировку лучей и управление ресурсами для повышения производительности.

Должен владеть:

методами работы с C++ в Unreal Engine, включая создание классов, функций, взаимодействие с игровыми объектами и управление памятью.

инструментами интеграции Blueprint и C++, используя BlueprintCallable, BlueprintImplementableEvent для взаимодействия игровых систем.

средствами разработки AI, анимаций и пользовательского интерфейса, включая Behavior Trees, AIController, BlendSpace, Event System и виджеты (Text, Image, ProgressBar).

## **2. Место дисциплины (модуля) в структуре ОПОП ВО**

Данная дисциплина (модуль) включена в раздел "Б1.В.ДВ.04.02 Дисциплины (модули)" основной профессиональной образовательной программы 09.03.04 "Программная инженерия (Индустрия разработки видеоигр)" и относится к дисциплинам по выбору части ОПОП ВО, формируемой участниками образовательных отношений.

Осваивается на 3 курсе в 5 семестре.

## **3. Объем дисциплины (модуля) в зачетных единицах с указанием количества часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся**

Общая трудоемкость дисциплины составляет 5 зачетных(ые) единиц(ы) на 180 часа(ов).

Контактная работа - 72 часа(ов), в том числе лекции - 0 часа(ов), практические занятия - 0 часа(ов), лабораторные работы - 72 часа(ов), контроль самостоятельной работы - 0 часа(ов).

Самостоятельная работа - 72 часа(ов).

Контроль (зачёт / экзамен) - 36 часа(ов).

Форма промежуточного контроля дисциплины: экзамен в 5 семестре.

**4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий**

**4.1 Структура и тематический план контактной и самостоятельной работы по дисциплине (модулю)**

N	Разделы дисциплины / модуля	Се-местр	Виды и часы контактной работы, их трудоемкость (в часах)						Само-стое-тель-ная ра-бота
			Лекции, всего	Лекции в эл. форме	Практические занятия, всего	Практические в эл. форме	Лабора-торные работы, всего	Лабора-торные в эл. форме	
1.	Тема 1. Тема 1. Основы работы с Unreal Engine	5	0	0	0	0	10	0	7
2.	Тема 2. Тема 2. Основы игрового взаимодействия	5	0	0	0	0	10	0	7
3.	Тема 3. Тема 3. Игроки и враги: механики взаимодействия	5	0	0	0	0	10	0	7
4.	Тема 4. Тема 4. Программирование на C++ в Unreal Engine	5	0	0	0	0	10	0	7
5.	Тема 5. Тема 5. Расширенные возможности C++ и Blueprint	5	0	0	0	0	10	0	8
6.	Тема 6. Тема 6. Продвинутые механики Unreal Engine	5	0	0	0	0	11	0	8
7.	Тема 7. Тема 7. Искусственный интеллект в Unreal Engine	5	0	0	0	0	11	0	28
	Итого		0	0	0	0	72	0	72

**4.2 Содержание дисциплины (модуля)**

**Тема 1. Тема 1. Основы работы с Unreal Engine**

Общая информация про игровые движки. Структура проекта на Unreal Engine. Blueprint как визуальный язык программирования. C++ в контексте Unreal Engine. Баланс между Blueprint и C++. Иерархия классов Unreal Engine: Actor, Pawn, Character, Player/AI Controller, GameMode.

Создание Blueprint-проекта с нуля. Интерфейс редактора Unreal Engine. Встроенное упрощённое моделирование (BSP brushes). Типы освещения. Transform. Установка готовых материалов. Создание класса на Blueprint. Actor и Pawn. SpringArm и Camera. GameMode и PlayerStart. Включение симуляции физики. Локальные и мировые координаты.

**Тема 2. Тема 2. Основы игрового взаимодействия**

Обработка ввода с внешних устройств - AxisMappings. Работами со "стандартными" событиями. Перемещение объекта с помощью физической симуляции и AddTorqueInRadians. Переменные. BoxCollision. Перезапуск уровня через OpenLevelByName. Литералы вместо переменных. PlayerCameraManager и тёмный экран. Создание материала. SphereCollision. Branch. Timeline - FloatTrack. Линейная интерполяция. Корневой компонент.

Типы функций: pure и impure. Создание функций. Локальные переменные. Создание макросов. ProjectileMovement. Event Hit. Импортирование игровых ресурсов. Создание акторов после запуска игры с помощью SpawnActorFromClass. Параметры переменных - InstanceEditable, категория, ValueRange.

Timeline - EventTrack Стандартная система работы с уроном - ApplyDamage и Event AnyDamage. Отладочный вывод с помощью PrintString. Создание библиотеки макросов. GetAllActorsOfClass. Массивы - создание, поиск нужного элемента. Категории функций. Сборка проекта в игру.

**Тема 3. Тема 3. Игроки и враги: механики взаимодействия**

Своя иерархия классов. Создание Character. SkeletalMesh. Создание Animation Blueprint с конечным автоматом. Создание библиотеки функций. Создание BlendSpace.

PlayerController - видимость курсора и вращение персонажа, GetHitResultUnderCursorByChannel. SkeletalMesh - создание сокета. Обработка ввода с внешних устройств - ActionMappings. Использование систем частиц. SoundCue - Random, Modulator. Виджеты - Text, Image, ProgressBar. Приведение типов. Отображение виджета, изменение данных виджета.

Искусственный интеллект - AIMoveTo и система навигации. Анимации - AnimMontage, AnimNotify. Использование родительской реализации обработки события и добавление специфики дочернего класса.

LineTraceByChannel. Использование таймеров. Преобразование классов. AnimationBlueprint - BlendPosesByBool.

Виджеты для неигрового персонажа.

#### **Тема 4. Тема 4. Программирование на C++ в Unreal Engine**

Создание проекта на C++ с нуля. Создание проекта на основе набора из магазина игровых ресурсов. Импортирование набора из магазина игровых ресурсов. Преобразование Blueprint-проекта в C++-проект. Создание класса на C++. Модификаторы доступа - public, protected, private. UPROPERTY - EditAnywhere, EditDefaultOnly, VisibleAnywhere, BlueprintReadWrite, BlueprintReadOnly. Сопоставление файлов с расширениями .cpp и .h с редактированием класса на Blueprint. Работа с примитивными типами - bool, int32, float. BeginPlay на C++.

Работа с FVector. Tick на C++. Комментирование исходного кода. Работа игры при разной частоте кадров с учётом и без учёта DeltaTime. UPROPERTY - Category.

Логирование на C++ - UE\_LOG.

Создание функций на C++ - константные функции, локальные переменные. Ранний выход из функций.

Редактирование StaticMesh - AddBoxSimplifiedCollision. Создание компонента на C++. UCLASS - BlueprintSpawnableComponent, Blueprintable. Указатели. Операторы ". " и "->". Как происходит сборка программы на C++, сборка игры на C++.

Создание TraceChannel. DrawDebugLine. Указатели и ссылки. SweepSingleByChannel.

#### **Тема 5. Тема 5. Расширенные возможности C++ и Blueprint**

UFUNCTION - BlueprintCallable. Нулевые указатели - как не допустить прерывание работы игры, как могут выглядеть связанные ошибки при запуске игры из среды разработки и напрямую из редактора движка. DrawDebugSphere. FHitResult - Location и ImpactPoint. UPhysicsHandleComponent - подбор и удержание физического объекта.

WakeAllRigidBodies. UPhysicsHandleComponent - прекращение удержания физического объекта. Массивы на C++ - создание, инициализация и получение элементов. Циклы while и for.

Разновидность цикла for - foreach. Метки акторов - добавление, проверка, удаление. Преобразование классов на C++. Прикрепление и открепление акторов.

Ограничение выбора класса на Blueprint через TSubclassOf. Работами со "стандартными" событиями на C++.

Стандартная система работы с уроном - ApplyDamage и OnTakeAnyDamage на C++.

Создание "нестандартных" виртуальных функций. Определение функции в заголовочном файле.

UFUNCTION - BlueprintImplementableEvent. Использование FTimerHandle не только как идентификатора таймера.

#### **Тема 6. Тема 6. Продвинутые механики Unreal Engine**

Работа с компонентами на C++ - объявление, инициализация, выстраивание иерархии. UPROPERTY как защита от ненужной сборки мусора. Обработка ввода с внешних устройств на C++ с помощью SetupPlayerInputComponent - BindAxis.

AddActorLocalOffset. UGameplayStatics - GetWorldDeltaSeconds. GetHitResultUnderCursor. StaticMesh и TraceComplex.

UGameplayStatics - GetPlayerPawn. Обработка ввода с внешних устройств на C++ с помощью SetupPlayerInputComponent - BindAction. Использование таймеров на C++.

Использование стандартных функций для настройки управления персонажем на C++.

Создание CameraShake.

Настройка BlendSpace под скорость движения в анимации. DrawDebugCamera. GetPlayerViewPoint.

Создание FPointDamageEvent. AAIController - SetFocus, MoveToActor.

APlayerController - GameHasEnded. TActorRange. Тернарный оператор.

Attenuation. Создание делегатов на C++.

#### **Тема 7. Тема 7. Искусственный интеллект в Unreal Engine**

AIController - LineOfSightTo, ClearFocus, StopMovement. Деревья поведения и blackboard: запуск дерева поведения; добавление ключей в blackboard; Sequence; SetValueAsVector; декораторы - BlackboardBasedCondition; ClearValue; создание задач дерева поведения на C++.

Деревья поведения: создание сервисов на C++; декораторы - Loop.

Деревья поведения и blackboard - SetValueAsObject. AmbientSound.

#### **5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)**

Самостоятельная работа обучающихся выполняется по заданию и при методическом руководстве преподавателя, но без его непосредственного участия. Самостоятельная работа подразделяется на самостоятельную работу на аудиторных занятиях и на внеаудиторную самостоятельную работу. Самостоятельная работа обучающихся включает как полностью самостоятельное освоение отдельных тем (разделов) дисциплины, так и проработку тем (разделов), осваиваемых во время аудиторной работы. Во время самостоятельной работы обучающиеся читают и конспектируют учебную, научную и справочную литературу, выполняют задания, направленные на закрепление знаний и отработку умений и навыков, готовятся к текущему и промежуточному контролю по дисциплине.

Организация самостоятельной работы обучающихся регламентируется нормативными документами, учебно-методической литературой и электронными образовательными ресурсами, включая:

Порядок организации и осуществления образовательной деятельности по образовательным программам высшего образования - программам бакалавриата, программам специалитета, программам магистратуры (утвержден приказом Министерства науки и высшего образования Российской Федерации от 6 апреля 2021 года №245)

Письмо Министерства образования Российской Федерации №14-55-996ин/15 от 27 ноября 2002 г. "Об активизации самостоятельной работы студентов высших учебных заведений"

Устав федерального государственного автономного образовательного учреждения "Казанский (Приволжский) федеральный университет"

Правила внутреннего распорядка федерального государственного автономного образовательного учреждения высшего профессионального образования "Казанский (Приволжский) федеральный университет"

Локальные нормативные акты Казанского (Приволжского) федерального университета

## **6. Фонд оценочных средств по дисциплине (модулю)**

Фонд оценочных средств по дисциплине (модулю) включает оценочные материалы, направленные на проверку освоения компетенций, в том числе знаний, умений и навыков. Фонд оценочных средств включает оценочные средства текущего контроля и оценочные средства промежуточной аттестации.

В фонде оценочных средств содержится следующая информация:

- соответствие компетенций планируемым результатам обучения по дисциплине (модулю);
- критерии оценивания сформированности компетенций;
- механизм формирования оценки по дисциплине (модулю);
- описание порядка применения и процедуры оценивания для каждого оценочного средства;
- критерии оценивания для каждого оценочного средства;
- содержание оценочных средств, включая требования, предъявляемые к действиям обучающихся, демонстрируемым результатам, задания различных типов.

Фонд оценочных средств по дисциплине находится в Приложении 1 к программе дисциплины (модулю).

## **7. Перечень литературы, необходимой для освоения дисциплины (модуля)**

Освоение дисциплины (модуля) предполагает изучение основной и дополнительной учебной литературы. Литература может быть доступна обучающимся в одном из двух вариантов (либо в обоих из них):

- в электронном виде - через электронные библиотечные системы на основании заключенных КФУ договоров с правообладателями;
- в печатном виде - в Научной библиотеке им. Н.И. Лобачевского. Обучающиеся получают учебную литературу на абонементе по читательским билетам в соответствии с правилами пользования Научной библиотекой.

Электронные издания доступны дистанционно из любой точки при введении обучающимся своего логина и пароля от личного кабинета в системе "Электронный университет". При использовании печатных изданий библиотечный фонд должен быть укомплектован ими из расчета не менее 0,5 экземпляра (для обучающихся по ФГОС 3++ - не менее 0,25 экземпляра) каждого из изданий основной литературы и не менее 0,25 экземпляра дополнительной литературы на каждого обучающегося из числа лиц, одновременно осваивающих данную дисциплину.

Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины (модуля), находится в Приложении 2 к рабочей программе дисциплины. Он подлежит обновлению при изменении условий договоров КФУ с правообладателями электронных изданий и при изменении комплектования фондов Научной библиотеки КФУ.

## **8. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)**

Multiplayer Network Compendium - <https://cedric-neukirchen.net/docs/category/multiplayer-network-compendium>

Документация Unreal Engine 4.27 -

[https://dev.epicgames.com/documentation/en-us/unreal-engine/unreal-engine-4-27-documentation?application\\_version=4.27](https://dev.epicgames.com/documentation/en-us/unreal-engine/unreal-engine-4-27-documentation?application_version=4.27)

Русскоязычный ресурс о компьютерных играх DTF - <https://dtf.ru/>

## **9. Методические указания для обучающихся по освоению дисциплины (модуля)**

Вид работ	Методические рекомендации
лабораторные работы	На лабораторных работах студент должен вести записи в специально отведенной для этого тетради. В случае, если студенту что-то непонятно, он может задать вопросы преподавателю. На лабораторных работах важно выполнять все задания преподавателя, внимательно следовать рекомендациям. Приветствуется активное участие в обсуждениях.
самостоятельная работа	В время самостоятельной работы студент должен повторять свои записи с занятий, выполнять домашние задания, формулировать вопросы, чтобы задать их на занятии, изучать рекомендованные источники. В освоении материала важную роль играет активная заинтересованность студента в изучаемом материале, поэтому приветствуется самостоятельное изучение дополнительных источников информации по теме, выполнение дополнительных заданий.
экзамен	Для подготовки к экзамену следует повторить свои записи, обратиться к основной и дополнительной литературе, рекомендованными источниками. Дополнительный самостоятельный поиск информации приветствуется, т. к. способствует глубокому пониманию изучаемого материала. Все вопросы студенты могут задать в течение курса и на консультации.

## **10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем (при необходимости)**

Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем, представлен в Приложении 3 к рабочей программе дисциплины (модуля).

## **11. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)**

Материально-техническое обеспечение образовательного процесса по дисциплине (модулю) включает в себя следующие компоненты:

Помещения для самостоятельной работы обучающихся, укомплектованные специализированной мебелью (столы и стулья) и оснащенные компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду КФУ.

Учебные аудитории для контактной работы с преподавателем, укомплектованные специализированной мебелью (столы и стулья).

Компьютер и принтер для распечатки раздаточных материалов.

## **12. Средства адаптации преподавания дисциплины к потребностям обучающихся инвалидов и лиц с ограниченными возможностями здоровья**

При необходимости в образовательном процессе применяются следующие методы и технологии, облегчающие восприятие информации обучающимися инвалидами и лицами с ограниченными возможностями здоровья:

- создание текстовой версии любого нетекстового контента для его возможного преобразования в альтернативные формы, удобные для различных пользователей;
- создание контента, который можно представить в различных видах без потери данных или структуры, предусмотреть возможность масштабирования текста и изображений без потери качества, предусмотреть доступность управления контентом с клавиатуры;
- создание возможностей для обучающихся воспринимать одну и ту же информацию из разных источников - например, так, чтобы лица с нарушениями слуха получали информацию визуально, с нарушениями зрения - аудиально;
- применение программных средств, обеспечивающих возможность освоения навыков и умений, формируемых дисциплиной, за счёт альтернативных способов, в том числе виртуальных лабораторий и симуляционных технологий;
- применение дистанционных образовательных технологий для передачи информации, организации различных форм интерактивной контактной работы обучающегося с преподавателем, в том числе вебинаров, которые могут быть использованы для проведения виртуальных лекций с возможностью взаимодействия всех участников дистанционного обучения, проведения семинаров, выступления с докладами и защиты выполненных работ, проведения тренингов, организации коллективной работы;
- применение дистанционных образовательных технологий для организации форм текущего и промежуточного контроля;
- увеличение продолжительности сдачи обучающимся инвалидом или лицом с ограниченными возможностями здоровья форм промежуточной аттестации по отношению к установленной продолжительности их сдачи;
- продолжительности сдачи зачёта или экзамена, проводимого в письменной форме, - не более чем на 90 минут;

- продолжительности подготовки обучающегося к ответу на зачёте или экзамене, проводимом в устной форме, - не более чем на 20 минут;
- продолжительности выступления обучающегося при защите курсовой работы - не более чем на 15 минут.

Программа составлена в соответствии с требованиями ФГОС ВО и учебным планом по направлению 09.03.04 "Программная инженерия" и профилю подготовки "Индустринг разработки видеоигр".

*Приложение 2  
к рабочей программе дисциплины (модуля)  
Б1.В.ДВ.04.02 Разработка высокопроизводительных игровых  
систем*

**Перечень литературы, необходимой для освоения дисциплины (модуля)**

Направление подготовки: 09.03.04 - Программная инженерия

Профиль подготовки: Индустрія разработки видеоигр

Квалификация выпускника: бакалавр

Форма обучения: очное

Язык обучения: русский

Год начала обучения по образовательной программе: 2025

**Основная литература:**

1. Белоцерковская, И. Е. Алгоритмизация. Введение в язык программирования C++: учебное пособие / Белоцерковская И. Е. , Галина Н. В. , Катаева Л. Ю. - Москва : Национальный Открытый Университет 'ИНТУИТ', 2016. - 197 с. - Текст : электронный // ЭБС 'Консультант студента' : [сайт]. - URL : <https://www.studentlibrary.ru/book/intuit026.html> (дата обращения: 10.12.2024). - Режим доступа : по подписке.
2. Фридман, А. Л. Язык программирования C++: учебное пособие / Фридман А. Л. - Москва : Национальный Открытый Университет 'ИНТУИТ', 2016. - 219 с. (Основы информационных технологий) - ISBN 5-9556-0017-5. - Текст : электронный // ЭБС 'Консультант студента' : [сайт]. - URL: <https://www.studentlibrary.ru/book/ISBN5955600175.html> (дата обращения: 10.12.2024). - Режим доступа : по подписке.
3. Фленов М. Искусство программирования игр на C++ / М. Фленов. - Санкт-Петербург : БХВ-Петербург, 2006. - 256 с. - ISBN 5-94157-832-6. - URL: <https://ibooks.ru/bookshelf/333679/reading> (дата обращения: 10.12.2024). - Текст: электронный.

**Дополнительная литература:**

1. Немцова, Т. И. Программирование на языке высокого уровня. Программирование на языке C++ : учебное пособие / Т.И. Немцова, С.Ю. Голова, А.И. Терентьев ; под ред. Л.Г. Гагариной. - Москва : ФОРУМ : ИНФРА-М, 2024. - 512 с. - (Среднее профессиональное образование). - ISBN 978-5-8199-0699-6. - Текст : электронный. - URL: <https://znanium.com/catalog/product/2083383> (дата обращения: 10.12.2024). - Режим доступа: по подписке.
2. Седжвик, Р. Алгоритмы на C++: учебное пособие / Седжвик Р. - Москва : Национальный Открытый Университет 'ИНТУИТ', 2016. - 1773 с. - Текст : электронный // ЭБС 'Консультант студента' : [сайт]. - URL : <https://www.studentlibrary.ru/book/intuit029.html> (дата обращения: 10.12.2024). - Режим доступа : по подписке.

*Приложение 3*

*к рабочей программе дисциплины (модуля)*

*Б1.В.ДВ.04.02 Разработка высокопроизводительных игровых  
систем*

**Перечень информационных технологий, используемых для освоения дисциплины (модуля), включая  
перечень программного обеспечения и информационных справочных систем**

Направление подготовки: 09.03.04 - Программная инженерия

Профиль подготовки: Индустрiya разработки видеоигр

Квалификация выпускника: бакалавр

Форма обучения: очное

Язык обучения: русский

Год начала обучения по образовательной программе: 2025

Освоение дисциплины (модуля) предполагает использование следующего программного обеспечения и информационно-справочных систем:

Операционная система Microsoft Windows 7 Профессиональная или Windows XP (Volume License)

Пакет офисного программного обеспечения Microsoft Office 365 или Microsoft Office Professional plus 2010

Браузер Mozilla Firefox

Браузер Google Chrome

Adobe Reader XI или Adobe Acrobat Reader DC

Kaspersky Endpoint Security для Windows