

**Автономная некоммерческая организация
дополнительного профессионального образования
«Центральный многопрофильный институт
профессиональной переподготовки и повышения квалификации»
АНО ДПО «ЦМИ»**

УТВЕРЖДАЮ
Директор АНО ДПО «ЦМИ»
А. В. Гриднева
«18» апреля 2023г.



**Дополнительная общеобразовательная
общеразвивающая программа
«Unreal Engine: от программирования на C++ до дизайна уровней в Blueprints»**

Направленность: техническая
уровень сложности «Начальный»

144 часа

**Общие данные о Дополнительной общеобразовательной программе
«Unreal Engine: от программирования на C++ до дизайна уровней в Blueprints»**

Об организации

Наименование поля	Допустимые значения поля	Значение поля
ИНН организации, осуществляющей образовательную деятельность	10 арабских цифр	2632108764
Наименование организации	строка	Автономная некоммерческая организация дополнительного профессионального образования «Центральный многопрофильный институт профессиональной переподготовки и повышения квалификации» АНО ДПО «ЦМИ»
Логотип организации	изображение в формате jpeg разрешением не менее 100x100 пиксель	
Ссылка на логотип организации	URL на изображение, находящееся в сети интернет	https://drive.google.com/file/d/1nobPZvgUGUtrHOVXT3yDlPdLZHXTB7nc/view?usp=drive_link
Контакты ответственного за программу (с указанием фамилии, имени, отчества).	строка от 5 до 255 символов	Гончарова Ирина Сергеевна

Контакты ответственного за программу. Должность	строка от 5 до 255 символов	РОП
Контакты ответственного за программу. Телефон	Формат +7(XXX)XXXXXXX	+7 9296469339
Контакты ответственного за программу. E-mail	строка	Umo_sk@mail.ru

Информация о программе

Наименование поля	Допустимые значения поля	Значение поля (примеры)
Название программы (курса)	строка	Unreal Engine: от программирования на C++ до дизайна уровней в Blueprints
Описание программы	строка не менее 1000 не более 5000 символов	Дополнительная общеобразовательная программа «Unreal Engine: от программирования на C++ до дизайна уровней в Blueprints» является неотъемлемой частью образовательной программы автономной некоммерческой организации дополнительного профессионального образования «Центральный многопрофильный институт профессиональной переподготовки и повышения квалификации» и дает возможность каждому ребенку получать дополнительное образование исходя из его интересов, склонностей и способностей, образовательных потребностей, осуществляемых за пределами федеральных государственных образовательных стандартов и федеральных государственных требований. По своему функциональному назначению программа является общеразвивающей и направлена на удовлетворение потребностей, обучающихся в интеллектуальном, нравственном совершенствовании, в организации их свободного времени. Направленность общеобразовательной программы «Unreal Engine: от программирования на C++ до дизайна уровней в Blueprints» техническая.
Аннотация (для размещения на маркетплейсе, понятное и привлекательное для Потенциальных получателей поддержки, включающее полное и	строка до 1000 символов	Изучение «Unreal Engine: от программирования на C++ до дизайна уровней в Blueprints» предоставляет детям уникальные возможности для разработки своих навыков в программировании, разработке игр. Этот курс предназначен для начинающих программистов, которые хотят изучить язык Lua и создать игры с помощью движка Roblox. Он поможет детям освоить основы программирования, работу с графикой, звуком и управлением игры, что даст им широкие возможности в будущей карьере в сфере разработки приложений. Требования к ученикам (уровень знаний):

содержательное описание Дополнительной общеобразовательной программы: 1) краткое описание Дополнительной общеобразовательной программы; 2) описание требований и рекомендаций для обучения по образовательной Дополнительной общеобразовательной программе; 3) краткое описание результатов обучения в свободной форме, включая описание практикоориентированного характера Дополнительной общеобразовательной программы)		● Базовые навыки владения компьютером (запуск приложений, создание папок, смена имени файла, скачивание картинок из интернета). Учащиеся должны владеть: Начальные знания английского языка и умение бегло печатать на английской раскладке. Результат обучения: ● Уверенное владение языком C++ и его основными конструкциями, такими как условные операторы, циклы и функции. ● Понимание принципов программирования игр и создание собственных игр на Unreal с помощью языка программирования C++. ● Умение создавать и настраивать графические элементы интерфейса, такие как кнопки, поля ввода и метки. ● Навыки обработки пользовательского ввода и отображения результатов на экране. ● Понимание принципов организации и структурирования кода, а также методологии разработки программных проектов. ● Уверенность в работе с командной строкой и управлении файлами в операционной системе.
---	--	---

Цель программы	строка не менее 100 символов	Цель обучения «Unreal Engine: от программирования на C++ до дизайна уровней в Blueprints» заключается в том, чтобы дать начинающим программистам возможность овладеть основными принципами программирования на языке C++, освоить создание игр с использованием Unreal Engine.
Актуальность	строка не менее 500 символов	Unreal Engine - это мощный игровой движок, который обладает множеством возможностей для создания высококачественных игр. Курс, который включает в себя программирование на C++ и дизайн уровней в Blueprints, представляет собой отличную возможность для тех, кто хочет освоить этот инструмент и начать создавать свои собственные игры. Программирование на C++ является одним из ключевых навыков, которые нужны для разработки игр на Unreal Engine. Большинство игровых движков написаны на C++, поэтому это знание является необходимым для создания эффективных игр. Курс позволяет участникам изучить основы C++, а также показывает, как его можно использовать для создания игровых функций. Blueprints - это визуальный язык программирования в Unreal Engine, который позволяет создавать игровые элементы без необходимости написания кода. Курс помогает участникам освоить Blueprint, чтобы создавать интерактивные игровые объекты и управлять ими. Это упрощает процесс создания игр и позволяет быстрее добиться результатов. Unreal Engine - это один из самых популярных игровых движков в мире. Он используется в создании многих крупных и успешных игр. Поэтому, знание Unreal Engine, включая умение программировать на C++ и разрабатывать дизайн уровней в Blueprint, является востребованным и ценным навыком в игровой индустрии. В целом, курс Unreal Engine от программирования на C++ до дизайна уровней в Blueprints актуален для тех, кто хочет начать свой путь в создании игр. Он помогает освоить необходимые навыки для работы с Unreal Engine, что открывает множество возможностей в игровой индустрии.

Дополнительная информация	строка	Педагогическая целесообразность программы заключается в привлечении учащихся к занятиям техническим творчеством, что способствует развитию логического мышления, творческих способностей и навыков решения задач программирования. Учащиеся вспоминают свои знания по основам алгоритмизации и программирования и на их основе, углубляя их, учатся составлять простые и сложные программы.
Формат обучения	значение из: очная форма без применения дистанцион- ных образователь- ных технологий; очная форма с применением дистанцион- ных образователь- ных технологий, в том числе, с применением средств электронного обучения	Очная форма с применением дистанционных образовательных технологий, в том числе, с применением средств электронного обучения

Уровень сложности	значение из: «Начальный» «Базовый» «Продвинутой»	Начальный
Срок освоения образовательной программы	строка, значение в ак.ч.	144 ак.ч.
Объем каждого модуля в ак.ч.	целое число	36
Объем часов в неделю в ак.ч.	целое число	4
Количество занятий	целое число	58
Направленность программы	строка	Современные языки программирования
Язык программирования	строка	Язык программирования C++
Дополнительная общеобразовательная программа не представлена для участия в иных федеральных проектах, направленных на дополнительное	строка, значения: «Не представлена»	Не представлена

образование граждан, кроме федерального проекта «Развитие кадрового потенциала ИТ-отрасли»		
Дополнительная общеобразовательная программа не была реализована до начала отбора и/или не реализовывается в период отбора на безвозмездной основе	строка, значения «Не реализована»	Не реализована
Категория обучающихся по программе	строка не менее 10 символов	Учащиеся 8 класса Учащиеся 9 класса Учащиеся 10 класса Учащиеся 11 класса Обучающиеся по программам среднего профессионального образования
Описание планируемых результатов обучения	строка не менее 10 символов	Сформировать у обучающихся базовые знания и умения языка программирования на C++, Сформировать умение применять программные средства для решения задач из различных предметных областей способствует развитию логического и комбинаторного мышления
Ссылка на лендинг Образовательной программы	строка не менее 10 символов	https://school.edu-sigma.ru/unrealengine

Ссылка на LMS	строка не менее 10 символов	https://odin.study/ru/
Страница обучения на курсе	строка не менее 10 символов	https://www.odin.study/ru/EducationalProgram/Info/7557

Аттестация

Промежуточная аттестация		
Количество академических часов	строка не менее 10 символов	4 часа текущая работа, включающая выполнение 4 практических задания
Формы контроля	строка не менее 10 символов	Текущий контроль. Практические задания. Для успешного завершения обучения по курсу необходимо, чтобы 50 % заданий в рамках текущей работы были зачтены (оценены преподавателем на 3, 4 или 5 баллов).
Диагностические инструменты	строка не менее 10 символов	Оценка полноты выполнения работы. Оценка способности оперировать полученными знаниями и умениями при решении практической задачи. Проверка результата запуска программного решения на языке C++.
Показатели и критерии оценивания	строка не менее 10 символов	Шкала оценивания: Нижнее значение - оценка «2»; Верхнее значение - оценка «5». Минимальный проходной балл для успешной сдачи - оценка «3» При оценке надо руководствоваться следующими критериями: 1) полнота и правильность выполнения заданий; 2) степень осознанности, понимания изученного; 3) оформление ответа.

		Оценка «5» ставится, если: 1) практическое задание выполнено без ошибок, полно изложен изученный материал; 2) учащийся понимает материал, может обосновать свои суждения и действия, хорошо применяет знания на практике, приводит необходимые примеры не только по учебному материалу, но и самостоятельно составленные; Оценка «4» ставится, если ученик выполнил практическое задание, удовлетворяющее тем же требованиям, что и для оценки «5», но допускает 1—2 ошибки, которые сам же исправляет, и 1—2 недочета в последовательности излагаемого материала. Оценка «3» ставится, если ученик освоил знания и понимает основы практического задания, но: 1) подготовил материал неполно и допускает неточности в последовательности выполнения заданий; 2) не может обосновать свои суждения и привести свои примеры; 3) допускает ошибки в оформлении. Оценка «2» ставится, если ученик обнаруживает незнание большей части изучаемого материала, допускает ошибки в написании кода, беспорядочно и неуверенно излагает материал. Оценка «2» отмечает такие недостатки в подготовке ученика, которые являются серьезным препятствием к успешному овладению последующим материалом.
Примеры заданий	строка не менее 10 символов	Модуль 1. Практическое задание № 1. Основы Unreal Engine Шаг 1: Создайте новый проект в Unreal Engine. Шаг 2: Добавьте в проект новый актер (Actor). Шаг 3: Откройте редактор Blueprint и создайте новый Blueprint класс на основе вашего актера. Шаг 4: Откройте созданный Blueprint и добавьте в него компонент Sphere (Сфера). Шаг 5: Создайте новый скрипт (C++ класс) и свяжите его с созданным Blueprint классом. Шаг 6: Добавьте в созданный скрипт метод, который будет изменять цвет вашей сферы. Шаг 7: Сохраните изменения и закройте редактор Blueprint. Шаг 8: Перейдите в режим игры (Play Mode) и убедитесь, что при нажатии на кнопку ваша сфера меняет цвет. Шаг 9: Загрузите файл в Odin

		Модуль 2. Практическое задание № 2. Программирование на C++ в Unreal Engine Шаг 1: Создайте новый проект в Unreal Engine. Шаг 2: Добавьте в проект новый актер (Actor). Шаг 3: Откройте редактор C++ классов и создайте новый класс на основе вашего актера. Шаг 4: Определите несколько переменных в созданном классе и проинициализируйте их. Шаг 5: Добавьте в класс метод, который будет изменять положение вашего актера. Шаг 6: Создайте объект вашего класса в игровом режиме (Play Mode) и вызовите созданный метод, чтобы переместить ваш актер в новую позицию. Шаг 7: Проверьте результат и убедитесь, что ваш актер изменил свое положение. Шаг 8: Загрузите файл в Odin Модуль 3. Практическое задание № 3 Продвинутое программирование в Unreal Engine Шаг 1: Создайте новый проект в Unreal Engine. Шаг 2: Откройте редактор C++ классов и создайте новый класс на основе вашего актера. Шаг 3: Определите несколько переменных в созданном классе и проинициализируйте их. Шаг 4: Добавьте в класс несколько методов, которые будут обрабатывать пользовательский ввод с клавиатуры или мыши. Шаг 5: Определите класс интерфейса пользователя (User Interface) и свяжите его с вашим классом актера. Шаг 6: Добавьте в интерфейс несколько кнопок и полей ввода для взаимодействия с пользователем. Шаг 7: Добавьте в класс актера методы, которые будут обрабатывать действия пользователя. Шаг 8: Создайте объект вашего класса в игровом режиме (Play Mode) и проверьте, что пользовательский ввод корректно обрабатывается и взаимодействие с пользователем работает правильно. Шаг 9: Добавьте в ваш класс методы для создания и управления несколькими актерами на сцене.
--	--	---

		Шаг 10: Создайте несколько объектов вашего класса и управляйте ими, используя пользовательский ввод и взаимодействие с интерфейсом пользователя. Шаг 11: Загрузите файл в Odin Модуль 4. Практическое задание № 4. Дизайн уровней в Blueprints Шаг 1: Создайте новый проект в Unreal Engine. Шаг 2: Откройте редактор уровней и создайте новый уровень. Шаг 3: Добавьте в уровень несколько статических мешей и акторов, чтобы создать окружение. Шаг 4: Откройте редактор Blueprints и создайте новый Blueprint класс на основе вашего игрового объекта (например, персонажа). Шаг 5: Определите несколько переменных в созданном классе и проинициализируйте их. Шаг 6: Добавьте в класс несколько методов, которые будут обрабатывать пользовательский ввод и управлять поведением вашего игрового объекта. Шаг 7: Свяжите ваш Blueprint класс с объектом на сцене. Шаг 8: Создайте несколько препятствий на уровне, которые игрок должен преодолеть, используя вашего персонажа. Шаг 9: Добавьте в уровень логику, которая будет определять условия прохождения уровня (например, достижение конечной точки или сбор определенного количества предметов). Шаг 10: Проверьте ваш уровень в игровом режиме (Play Mode) и убедитесь, что все логические условия работают правильно, а персонаж может пройти уровень. Шаг 11: Загрузите файл в Odin
Шкала оценивания, нижнее значение	строка не менее 10 символов	2
Шкала оценивания, верхнее значение	строка не менее 10 символов	5

Шкала оценивания, минимальный проходной балл	строка не менее 10 символов	3
--	--------------------------------	---

Преподаватели

ФИО	Наименование основного места работы	Должность	Высшее образование или среднее профессиональное образование по направлению «Образование и педагогические науки»	Высшее образование или среднее профессиональное образование по иному направлению соответствующим направленности ДОП	Ссылка на веб-страницы с портфолио	Информация о курсах повышения квалификации по профилю преподаваемой дисциплины (за последние 3 года)	Пройдена промежуточная аттестация не менее чем за два года обучения по образовательным программам высшего образования по специальностям и направлениям подготовки, соответствующим направленности ДОП	Отметка о полученном согласии на обработку персональных данных
строка от 2 до 100 символов	строка от 2 до 255 символов.	строка от 2 до 255 символов	да/нет	да/нет	строка		да/нет	да/нет
Максимов Егор Васильевич	Fog- stream	Преподаватель Python	нет	да	https://docs.google.com/document/d/1W	нет	нет	да

					hqIIOG2b4S_fAczP9z-kZSdTAdigj98/edit#			
Борисова Анна Игоревна	КГБ ПОУ «Хабаровский дорожно-строительный техникум»	Преподаватель информатики	да	да	https://docs.google.com/document/d/1EOKNAckojHxXzeprx1PMXRqCqhY6QCI7/edit#	нет	нет	да

I ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Дополнительная общеобразовательная программа «Unreal Engine: от программирования на C++ до дизайна уровней в Blueprints» является неотъемлемой частью образовательной программы Автономной некоммерческой организации дополнительного профессионального образования «Центральный многопрофильный институт профессиональной переподготовки и повышения квалификации» АНО ДПО «ЦМИ» и дает возможность каждому ребенку получать дополнительное образование исходя из его интересов, склонностей и способностей, образовательных потребностей, осуществляемых за пределами федеральных государственных образовательных стандартов и федеральных государственных требований.

По своему функциональному назначению программа является общеразвивающей и направлена на удовлетворение потребностей, обучающихся в интеллектуальном, нравственном совершенствовании, в организации их свободного времени.

Направленность общеобразовательной программы «Unreal Engine: от программирования на C++ до дизайна уровней в Blueprints» техническая.

Актуальность и педагогическая целесообразность программы

Программа обучения Unreal Engine, которая включает в себя программирование на C++ и дизайн уровней в Blueprints, имеет как актуальность, так и педагогическую целесообразность. С точки зрения актуальности, Unreal Engine - это один из наиболее популярных игровых движков в мире, который используется для создания игр различных жанров и платформ. Он является мощным инструментом для разработки игр, и знание Unreal Engine, включая программирование на C++ и дизайн уровней в Blueprints, является востребованным и ценным навыком в игровой индустрии. Более того, Unreal Engine обновляется и улучшается постоянно, что делает знание этого движка еще более актуальным.

С точки зрения педагогической целесообразности, программа обучения Unreal Engine является эффективным способом для начинающих и опытных разработчиков игр узнать, как создавать игры на основе Unreal Engine. Курс, который включает в себя программирование на C++ и дизайн уровней в Blueprints, позволяет участникам изучать движок на разных уровнях сложности и приобретать навыки в соответствии с их уровнем подготовки. Кроме того, использование различных подходов к обучению, таких как практические занятия, демонстрации и объяснения, обеспечивает более глубокое и понятное усвоение материала.

В целом, программа обучения Unreal Engine, которая включает в себя программирование на C++ и дизайн уровней в Blueprints, является актуальной и педагогически целесообразной. Она позволяет участникам изучить Unreal Engine на разных уровнях сложности и приобрести навыки, которые могут быть использованы в дальнейшей работе в игровой индустрии.

Цель и основные задачи программы: формирование первичных компетенций в области информационных технологий, создания игр, навыков исследовательской и изобретательской деятельности, проектной и командной работы.

Основные задачи:

Образовательные:

- Овладение основными концепциями и принципами программирования на языке C++.
- Приобретение навыков работы с Unreal Engine для разработки компьютерных игр.
- Освоение принципов работы с графическим интерфейсом.
- Понимание методов и приемов работы с графикой, звуком и управлением игровым процессом.
- Разработка собственных игр и приложений с использованием полученных знаний и навыков.

Воспитательные:

- воспитать мотивацию учащихся к изобретательству, созданию собственных игр;
- привить стремление к получению качественного законченного результата в проектной деятельности;
- формировать правильное восприятие системы ценностей, принципов, правил информационного общества;
- формировать потребность в самостоятельном приобретении и применении знаний, потребность к постоянному саморазвитию;
- воспитывать социально-значимые качества личности человека: ответственность, коммуникабельность, добросовестность, взаимопомощь, доброжелательность.

Развивающие:

- способствовать развитию творческих способностей учащихся, познавательных интересов, развитию индивидуальности и самореализации;
- расширять технологические навыки при подготовке различных информационных материалов;
- развивать познавательные способности ребенка, память, внимание, пространственное мышление, аккуратность и изобретательность при выполнении учебных проектов;
- формировать творческий подход к поставленной задаче;
- развивать навыки инженерного мышления, умения работать как по предложенным инструкциям, так и находить свои собственные пути решения поставленных задач;
- формировать навыки рефлексивной деятельности.

Педагогические принципы, определяющие теоретические подходы к построению образовательного процесса

Реализация программы «Unreal Engine: от программирования на C++ до дизайна уровней в Blueprints» основывается на принципах научности, последовательности, системности, связи теории с практикой, доступности.

В целях раскрытия педагогического и развивающего потенциала учебно-воспитательного процесса по программе акцент в ней делается на следующих принципах:

Принцип прочности предполагает стремление к тому, чтобы приобретенные знания, умения и навыки обучающихся были прочно закреплены, в первую очередь, практически.

Для этого необходимо их участие в соревнованиях, где полученные умения и навыки можно проверить на практике, в выездных мероприятиях, а также постоянное поддержание интереса обучающихся к занятиям.

Принцип проектности предполагает последовательную ориентацию всей деятельности педагога на подготовку и выведение ребенка в самостоятельное проектное действие, осуществляемое в логике замысел – реализация – рефлексия. В ходе проектирования перед человеком всегда стоит задача представить себе еще не существующее, но то, что он хочет, чтобы появилось в результате его активности.

Принцип активности предполагает необходимость строить учебный процесс таким образом, чтобы учащиеся не просто воспринимали предлагаемый материал, но и стремились закрепить полученные знания, умения и навыки, анализировали ошибки и достижения свои и товарищей, самостоятельно решали поставленные задачи.

Основные характеристики образовательного процесса

Уровень освоения программы начальный, что предполагает освоение обучающимися специализированных знаний, обеспечение трансляции общей и целостной картины тематического содержания программы.

Форма обучения: очная форма с применением дистанционных образовательных технологий, в том числе, с применением средств электронного обучения

Режим работы: занятия проводятся 2-3 раза в неделю по 2 академических часа, длительность одного академического часа – 45 минут.

Продолжительность образовательного процесса: начало занятий 01 октября 2023 г., завершение 31 мая 2024 г.

Объем учебных часов по программе 144 ак.часа.

Ожидаемые результаты освоения программы

Личностные:

- развита способность переносить (выдерживать) известные нагрузки в течение определенного времени, преодолевать трудности;
- умеет контролировать свои поступки;
- умеет воспринимать общие дела как свои собственные.

Метапредметные:

- развито умение представлять результаты своей работы окружающим, аргументировать свою позицию;
- развита способность правильно организовывать рабочее место и время для достижения поставленных целей.
- сформировано у учащихся умение работать в команде.

Предметные:

- сформированы практические и теоретические знания в области устройства и функционирования современных платформ;
- изучены основы алгоритмизации, построения алгоритмов;
- научился формулировать и анализировать алгоритмы;
- изучить основы программирования на языке C++
- научиться разрабатывать игры на Unreal Engine с помощью языка программирования C++.

Данная программа адаптирована для занятий с учащимися среднего и старшего возраста, с различной степенью подготовки, включает теоретические и практические задания.

Входные требования к обучающимся: тестирование, базовое владение персональным компьютером.

Для занятий -наличие ПК, камеры, микрофона, стабильного доступа в интернет.

Календарный учебный график формируется непосредственно при реализации общеобразовательной программы «Unreal Engine: от программирования на C++ до дизайна уровней в Blueprints».

Первый модуль с 01.10.2023 - 30.11.2023

Второй модуль: с 01.12.2023 - 31.01.2024

Третий модуль: с 01.02.2024 - 31.03.2024

Четвертый модуль: с 01.04.2024 - 31.05.2024

Занятия проводятся 2-3 раза в неделю по 2 академических часа, длительность одного академического часа – 45 минут.

СОДЕРЖАНИЕ ОБЩЕРАЗВИВАЮЩЕЙ ПРОГРАММЫ

СОДЕРЖАНИЕ УЧЕБНОГО (ТЕМАТИЧЕСКОГО) ПЛАНА

«Unreal Engine: от программирования на C++ до дизайна уровней в Blueprints»

№	Наименование модулей (курсов)	Общая трудоемкость, ч.	Всего контактных часов		Контактные часы		СРС, ч.	Формы контроля
п/п			Общее количество Лекций и ПЗ	ДЗ	Лекции	Практические занятия		
I	Модуль 1: Основы Unreal Engine	36	30	1	9	21	6	Зачет
1.1	Тема 1.1 Введение в Unreal Engine	1	1		1			
1.2	Тема 1.2 Установка и настройка Unreal Engine	1	1		1			
1.3	Тема 1.3 Основы работы с редактором Unreal Engine	1	1		1			
1.4	Тема 1.4 Создание простого уровня	7	6		1	5	1	
1.5	Тема 1.5 Работа с материалами и освещением	7	6		1	5	1	
1.6	Тема 1.6 Создание первого игрового объекта	4	3		1	2	1	

1.7	Тема 1.7 Работа со звуком и анимацией	5	4		1	3	1	
1.8	Тема 1.8 Импорт и использование активов	5	4		1	3	1	
1.9	Тема 1.9 Экспорт и публикация игры	5	4	1	1	3	1	Зачет
2	Модуль 2: Программирование на C++ в Unreal Engine	36	30	1	9	21	6	Зачет
2.1	Тема 2.1 Введение в программирование на C++ в Unreal Engine	2	2		1	1		
2.2	Тема 2.2 Основы языка C++	3	3		1	2		
2.3	Тема 2.3 Работа с классами и объектами	4	3		1	2	1	
2.4	Тема 2.4 Использование наследования	4	3		1	2	1	
2.5	Тема 2.5 Операторы и выражения в C++	4	3		1	2	1	
2.6	Тема 2.6 Работа с массивами и указателями	4	4		1	3		
2.7	Тема 2.7 Использование шаблонов и контейнеров	3	3		1	2		
2.8	Тема 2.8 Работа с файлами и потоками ввода-вывода	4	3		1	2	1	
2.9	Тема 2.9 Создание простого игрового компонента на C++	4	3	1	1	2	1	
2.10	Тема 2.10 Взаимодействие с графическим интерфейсом пользователя в Unreal Engine: создание элементов управления,	4	3			3	1	Зачет

	обработка событий.							
3	Модуль 3: Продвинутое программирование в Unreal Engine	36	30	1	10	20	6	Зачет
3.1	Тема 3.1 Архитектура Unreal Engine	4	3		1	2	1	
3.2	Тема 3.2 Работа с игровыми объектами и компонентами	4	3		1	2	1	
3.3	Тема 3.3 Работа с анимацией и физикой	4	3		1	2	1	
3.4	Тема 3.4 Работа с интерфейсом пользователя	4	3	1	1	2	1	Зачет
3.5	Тема 3.5 Работа с сетевыми играми	3	3		1	2		
3.6	Тема 3.6 Оптимизация производительности	4	4		1	3		
3.7	Тема 3.7 Работа с внешними библиотеками и плагинами	4	4		1	3		
3.8	Тема 3.8 Использование асинхронных задач и потоков	4	3		1	2	1	
3.9	Тема 3.9 Разработка игры на C++ с использованием Unreal Engine	5	4		2	2	1	
4	Модуль 4: Дизайн уровней в Blueprints	36	30	1	9	21	6	Зачет
4.1	Тема 4.1 Введение в Blueprints	5	4		1	3	1	
4.2	Тема 4.2 Основы работы с Blueprints	4	4		2	2		

4.3	Тема 4.3 Создание простого игрового объекта в Blueprints	5	4		1	3	1	
4.4	Тема 4.4 Использование переменных и функций в Blueprints	4	3		1	2	1	
4.5	Тема 4.5 Создание пользовательских функций и узлов	5	4		1	3	1	
4.6	Тема 4.6 Работа с событиями и триггерами, анимацией и физикой в Blueprints. Создание и настройка UI в Blueprints	6	5		2	3	1	
4.7	Тема 4.7 Разработка игры в Unreal Engine с использованием Blueprints	5	4	1	1	3	1	Зачет
	Консультация	2	2			2		
	Итого	144	120	4	37	83	24	

II. ОЦЕНКА КАЧЕСТВА ОСВОЕНИЯ ПРОГРАММЫ (формы аттестации, оценочные и методические материалы)

В курсе будут оцениваться:

текущая работа, включающая выполнение 4 практических задания.

Задания будут оцениваться по пятибалльной шкале :

Шкала оценивания:

Нижнее значение 2

Верхнее значение 5

Минимальный проходной балл для успешной сдачи 3

При оценке следует руководствоваться следующими критериями:

- 1) полнота и правильность выполнения заданий;
- 2) степень осознанности, понимания изученного;
- 3) оформление ответа.

Оценка «5» ставится, если:

1. Практическое задание выполнено без ошибок, полно изложен изученный материал;
2. учащийся понимает материал, может обосновать свои суждения и действия, хорошо применяет знания на практике, способен привести необходимые примеры не только по учебному материалу, но и составленные самостоятельно.

Оценка «4» ставится, если ученик выполнил практическое задание, удовлетворяющее тем же требованиям, что и для оценки «5», но допустил 1-2 ошибки, которые сам же исправляет, и 1-2 недочета в последовательности излагаемого материала.

Оценка «3» ставится, если ученик освоил знания и понимает основы практического задания, но:

- 1) подготовил материал неполно и допускает неточности в последовательности выполнения заданий;
- 2) не смог обосновать свои суждения и привести свои примеры;
- 3) допустил ошибки в оформлении.

Оценка «2» ставится, если ученик обнаруживает незнание большей части изучаемого материала, допускает ошибки в написании кода, беспорядочно и неуверенно излагает материал.

Оценка «2» отмечает такие недостатки в подготовке ученика, которые являются серьезным препятствием к успешному овладению последующим материалом.

Для успешного завершения обучения по курсу необходимо, чтобы 50 % заданий в рамках текущей работы были зачтены (оценены преподавателем на 3, 4 или 5 баллов).

В таблице представлен перечень заданий, которые будут проводиться на практике в режиме онлайн с преподавателем в период обучения на курсе.

№ п/п	Мероприятия промежуточного контроля	Количество баллов
1.	Практическое задание № 1. Основы Unreal Engine	2-5
2.	Практическое задание № 2. Программирование на C++ в Unreal Engine	2-5
3.	Практическое задание № 3. Продвинутое программирование в Unreal Engine	2-5
4.	Практическое задание № 4. Дизайн уровней в Blueprints	2-5

Оценочные материалы для текущего контроля успеваемости:

Модуль 1. Практическое задание № 1. Основы Unreal Engine

Шаг 1: Создайте новый проект в Unreal Engine.

Шаг 2: Добавьте в проект новый актер (Actor).

Шаг 3: Откройте редактор Blueprint и создайте новый Blueprint класс на основе вашего актера.

Шаг 4: Откройте созданный Blueprint и добавьте в него компонент Sphere (Сфера).

Шаг 5: Создайте новый скрипт (C++ класс) и свяжите его с созданным Blueprint классом.

Шаг 6: Добавьте в созданный скрипт метод, который будет изменять цвет вашей сферы.

Шаг 7: Сохраните изменения и закройте редактор Blueprint.

Шаг 8: Перейдите в режим игры (Play Mode) и убедитесь, что при нажатии на кнопку ваша сфера меняет цвет.

Шаг 9: Загрузите файл в Odin

Модуль 2. Практическое задание № 2. Программирование на C++ в Unreal Engine

Шаг 1: Создайте новый проект в Unreal Engine.

Шаг 2: Добавьте в проект новый актер (Actor).

Шаг 3: Откройте редактор C++ классов и создайте новый класс на основе вашего актера.

Шаг 4: Определите несколько переменных в созданном классе и проинициализируйте их.

Шаг 5: Добавьте в класс метод, который будет изменять положение вашего актера.

Шаг 6: Создайте объект вашего класса в игровом режиме (Play Mode) и вызовите созданный метод, чтобы переместить ваш актер в новую позицию.

Шаг 7: Проверьте результат и убедитесь, что ваш актер изменил свое положение.

Шаг 8: Загрузите файл в Odin

Модуль 3. Практическое задание № 3 Продвинутое программирование в Unreal Engine

Шаг 1: Создайте новый проект в Unreal Engine.

Шаг 2: Откройте редактор C++ классов и создайте новый класс на основе вашего актера.

Шаг 3: Определите несколько переменных в созданном классе и проинициализируйте их.

Шаг 4: Добавьте в класс несколько методов, которые будут обрабатывать пользовательский ввод с клавиатуры или мыши.

Шаг 5: Определите класс интерфейса пользователя (User Interface) и свяжите его с вашим классом актера.

Шаг 6: Добавьте в интерфейс несколько кнопок и полей ввода для взаимодействия с пользователем.

Шаг 7: Добавьте в класс актера методы, которые будут обрабатывать действия пользователя.

Шаг 8: Создайте объект вашего класса в игровом режиме (Play Mode) и проверьте, что пользовательский ввод корректно обрабатывается и взаимодействие с пользователем работает правильно.

Шаг 9: Добавьте в ваш класс методы для создания и управления несколькими актерами на сцене.

Шаг 10: Создайте несколько объектов вашего класса и управляйте ими, используя пользовательский ввод и взаимодействие с интерфейсом пользователя.

Шаг 11: Загрузите файл в Odin

Модуль 4. Практическое задание № 4. Дизайн уровней в Blueprints

Шаг 1: Создайте новый проект в Unreal Engine.

Шаг 2: Откройте редактор уровней и создайте новый уровень.

Шаг 3: Добавьте в уровень несколько статических мешей и акторов, чтобы создать окружение.

Шаг 4: Откройте редактор Blueprints и создайте новый Blueprint класс на основе вашего игрового объекта (например, персонажа).

Шаг 5: Определите несколько переменных в созданном классе и проинициализируйте их.

Шаг 6: Добавьте в класс несколько методов, которые будут обрабатывать пользовательский ввод и управлять поведением вашего игрового объекта.

Шаг 7: Свяжите ваш Blueprint класс с объектом на сцене.

Шаг 8: Создайте несколько препятствий на уровне, которые игрок должен преодолеть, используя вашего персонажа.

Шаг 9: Добавьте в уровень логику, которая будет определять условия прохождения уровня (например, достижение конечной точки или сбор определенного количества предметов).

Шаг 10: Проверьте ваш уровень в игровом режиме (Play Mode) и убедитесь, что все логические условия работают правильно, а персонаж может пройти уровень.

Шаг 11: Загрузите файл в Odin

III. КАДРОВЫЕ УСЛОВИЯ

Руководитель программы:

Сергеев Олег Владимирович, Российский государственный социальный университет, Москва Факультет: Информационные технологии (бакалавриат) Специальность: Педагогическое образование (информатика) Сентябрь 2011 - Июль 2015 Российский государственный социальный университет, Москва Факультет: Информационные технологии (магистратура) Специальность: Педагогическое образование (информатика) Сентябрь 2016 - Июль 2019

Стаж работы – более 7 лет.

Сертификаты/ курсы:

Разработка веб-приложений с использованием Javascript ФГАОУ ВО "Национальный исследовательский университет "Высшая школа экономики". Московский институт электроники и математики им. А. Н. Тихонова 2019 год.

Программирование веб-сайтов и веб-приложений на HTML и CSS ФГАОУ ВО "Национальный исследовательский университет "Высшая школа экономики". Московский институт электроники и математики им. А. Н. Тихонова 2018 год.

Методика и использование 3d-прототипирования и моделирования на учебных занятиях в инженерном классе Академия «Просвещение» 2017 год.

«Подготовка технических специалистов, оказывающих информационно-техническую помощь руководителю и организаторам пункта проведения экзамена при проведении государственной итоговой аттестации по образовательным программам основного общего образования» ГАОУ ДПО МЦКО 2017 год.

"ИК технологии при обучении робототехнике" Городской методический центр Департамента образования города Москвы 2016 год Дизайнер-верстальщик РГСУ 2013 год

«Методика обучения технологии программирования объектов дополненной реальности на Unity 3D». Городской Методический Центр 2023 год.

III. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ МОДУЛЯ (организационно-педагогические)

Материально-технические условия реализации программы:
Материально-технические условия реализации программы

Обучение по программе реализовано в режиме синхронно-асинхронной работы слушателей в электронной среде. Для проведения синхронных занятий применяется программа видеоконференцсвязи Zoom. Дополнительно для организации работы слушателей могут использоваться Google-сервисы. Асинхронная работа слушателей реализуется на базе электронного курса на платформе электронного обучения LMS Odin.

Требования к оборудованию

Сеть: скорость соединения от 2 Мб/с.

Оборудование для синхронных занятий: персональный компьютер (рекомендуется) / мобильный телефон / планшет; наушники, микрофон и камера (обязательно).

Для работы на платформе электронного обучения LMS Odin рекомендуется использовать персональный компьютер.

Для успешной реализации Программы необходимо, чтобы рабочее место обучающегося и преподавателя включали в себя:

- компьютеры, обеспечивающие возможность работы с мультимедийным контентом: воспроизведение видеоизображений, качественный стереозвук в наушниках, речевой ввод с микрофона и др.

периферийное оборудование:

- принтер (черно/белой печати, формата А4);
- устройства для ввода визуальной информации (сканер, цифровой фотоаппарат, web-камера и пр.);

- устройства создания графической информации (графический планшет), использующиеся для создания и редактирования графических объектов, ввода рукописного текста;
- акустические колонки;
- оборудование, обеспечивающее подключение к сети Интернет (комплект оборудования для подключения к сети Интернет, сервер).

программное обеспечение компьютера:

- операционная система семейства MacOS или Windows;
- программа Unreal Engine,
- файловый менеджер (в составе операционной системы или др.);
- почтовый клиент (в составе операционных систем или др.);
- браузер (в составе операционных систем или др.);
- мультимедиа проигрыватель (в составе операционной системы или др.);
- антивирусная программа;
- программа-архиватор;
- программа-переводчик;
- программа интерактивного общения;
- текстовый редактор;
- растровый графический редактор;
- звуковой редактор;
- редактор Web-страниц.

Учебно-методическое и информационное обеспечение программы:

Методические рекомендации и пособия по изучению курса.

Программа реализуется в формате смешанного обучения. Синхронные занятия включают интерактивные лекции и практические занятия, предусматривающие групповую и индивидуальную работу слушателей. Для организации асинхронной работы слушателей используются записи синхронных занятий, презентации, конспекты лекций, практические задания и тестовые вопросы, размещаемые в электронном курсе на платформе электронного обучения LMS Odin.

Содержание комплекта учебно-методических материалов.

Для изучения программы используется электронный курс в LMS Odin. Электронный курс включает информационно-организационные материалы по программе (учебно-тематический план, календарный график работы по программе, информацию о преподавателях программы, чат для объявлений и вопросов, сведения о результатах обучения), набор видеолекций, презентации к лекциям, набор ссылок на внешние ресурсы, систему заданий с инструкциями, списки основной и дополнительной литературы.

Список литературы

Литература для педагога

1. К. Виллет. Unreal Engine. Профессиональная разработка игр. 2016.

2. Х. Дикман. Разработка игр в Unreal Engine 4. 2017.
3. А. Адамович, А. Сергиенко. Unreal Engine 4: игровая разработка в примерах. 2018.
4. Б. Баранов, Ю. Зуев. Unreal Engine 4: программирование и разработка игр. 2018.
5. Д. Шейд. Unreal Engine 4 для профессионалов. 2016.
6. Д. Рид. Разработка компьютерных игр в Unreal Engine 4. 2017.
7. М. Миткевич. Unreal Engine 4 для начинающих. 2018.
8. Ш. Паттерсон. Unreal Engine 4 для дизайнеров уровней. 2017.
9. А. Рязанов. Unreal Engine 4: разработка приложений виртуальной реальности. 2019.
10. М. Пирогов. Unreal Engine 4 для мобильных устройств. 2019

Компетенции

Наименование компетенции	Тип компетенции	Знания, соответствующие компетенции	Умения, соответствующие компетенции	Владение инструментами, соответствующие компетенции
строка, не менее 10 символов	строка, значение из «общекультурные», «общепрофессиональные», «профессиональные»	строка не менее 50 символов, перечень знаний	строка не менее 50 символов, перечень умений	строка не менее 50 символов, перечень инструментов
ПК-1 Создание программ на языке Python	общепрофессиональные	- Принципы работы информационных технологий - Принципы работы алгоритмов - Существующие структуры данных - Принципы написания программ - Работа с командной строкой - Подключение библиотек	- Устанавливать Unreal Engine на ПК - Писать циклы, условия, функции на C++ - Подключать сторонние библиотеки	- Unreal Engine - знание основного интерфейса, функций и возможностей редактора Unreal Engine, а также умение создавать и настраивать сцены, объекты, компоненты и ресурсы. - C++ - знание языка программирования C++ и его особенностей в контексте разработки игр на движке Unreal Engine. - Графический редактор - знание графического редактора, который

				используется для создания и редактирования текстур, моделей, спрайтов и других графических элементов для использования в Unreal Engine. -Анимация - знание принципов анимации и умение создавать анимационные контроллеры и анимации для объектов и персонажей. -Физика - знание физической системы в Unreal Engine и умение настраивать физические свойства объектов и их взаимодействия. -Написание скриптов - умение создавать скрипты на C++ для реализации логики игры и управления поведением объектов. -Оптимизация - знание методов и инструментов оптимизации производительности игры в Unreal Engine.
--	--	--	--	--

				-Работа с аудио и видео - умение работать с аудио и видео файлами и настраивать их воспроизведение в Unreal Engine. -Работа с сетью - умение создавать многопользовательские игры и работать с сетевыми протоколами и API для реализации многопользовательского взаимодействия.
--	--	--	--	---

Модули

Наименование поля	Допустимые значения полей	Значение полей	Значение полей	Значение полей	Значение полей
Порядковый номер модуля	целое число	1	2	3	4
Название модуля	1.	Модуль 1: Основы Unreal Engine	Модуль 2: Программирование на C++ в Unreal Engine	Модуль 3: Продвинутое программирование в Unreal Engine	Модуль 4: Дизайн уровней в Blueprints
Описание модуля	2.	Модуль 1. В рамках прохождения данного модуля будут рассмотрены темы: Тема 1.1 Введение в Unreal Engine Тема 1.2 Установка и настройка Unreal Engine Тема 1.3 Основы работы с редактором Unreal Engine Тема 1.4 Создание простого уровня Тема 1.5 Работа с материалами и освещением Тема 1.6 Создание первого игрового объекта Тема 1.7 Работа со звуком и анимацией Тема 1.8 Импорт и использование активов	Модуль 2. В рамках прохождения данного модуля будут рассмотрены темы: Тема 2.1 Введение в программирование на C++ в Unreal Engine Тема 2.2 Основы языка C++ Тема 2.3 Работа с классами и объектами Тема 2.4 Использование наследования Тема 2.5 Операторы и выражения в C++ Тема 2.6 Работа с массивами и указателями	Модуль 3. В рамках прохождения данного модуля будут рассмотрены темы: Тема 3.1 Архитектура Unreal Engine Тема 3.2 Работа с игровыми объектами и компонентами Тема 3.3 Работа с анимацией и физикой Тема 3.4 Работа с интерфейсом пользователя Тема 3.5 Работа с сетевыми играми Тема 3.6 Оптимизация производительности	Модуль 4. В рамках прохождения данного модуля будут рассмотрены темы: Тема 4.1 Введение в Blueprints Тема 4.2 Основы работы с Blueprints Тема 4.3 Создание простого игрового объекта в Blueprints Тема 4.4 Использование переменных и функций в Blueprints Тема 4.5 Создание пользовательских функций и узлов

		Тема 1.9 Экспорт и публикация игры	Тема 2.7 Использование шаблонов и контейнеров Тема 2.8 Работа с файлами и потоками ввода-вывода Тема 2.9 Создание простого игрового компонента на C++ Тема 2.10 Взаимодействие с графическим интерфейсом пользователя в Unreal Engine: создание элементов управления, обработка событий.	Тема 3.7 Работа с внешними библиотеками и плагинами Тема 3.8 Использование асинхронных задач и потоков Тема 3.9 Разработка игры на C++ с использованием Unreal Engine	Тема 4.6 Работа со событиями и триггерами, анимацией и физикой в Blueprints. Создание и настройка UI в Blueprints Тема 4.7 Разработка игры в Unreal Engine с использованием Blueprints
Аттестация по итогам модуля. Количество ак. часов	целое число	1	1	1	1
Аттестация по итогам модуля. Формы контроля	строка не менее 4 символов	Зачет	Зачет	Зачет	Зачет

Аттестация по итогам модуля. Диагностические инструменты	строка не менее 10 символов	Практическое задание № 1. Модуль 1: Основы Unreal Engine	Практическое задание № 2. Модуль 2: Программирование на C++ в Unreal Engine	Практическое задание № 3. Модуль 3: Продвинутое программирование в Unreal Engine	Практическое задание № 4. Модуль 4: Дизайн уровней в Blueprints
Аттестация по итогам модуля. Показатели и критерии оценивания	строка не менее 50 символов	В ходе реализации программы предусмотрены следующие виды оценочных мероприятий: - практические задания и упражнения в период синхронной работы; - индивидуальные задания, в том числе со взаимной проверкой (комментированием, рецензированием).			
Аттестация по итогам модуля. Шкала оценивания, нижнее значение	целое число	2-5	2-5	2-5	2-5
Аттестация по итогам модуля. Шкала оценивания, верхнее значение	целое число	2-5	2-5	2-5	2-5
Аттестация по итогам модуля. Шкала оценивания, минимальный проходной балл	целое число в диапазоне шкалы	3	3	3	3

для успешной сдачи					
--------------------	--	--	--	--	--

РАБОЧАЯ ПРОГРАММА

«Unreal Engine: от программирования на C++ до дизайна уровней в Blueprints»

АННОТАЦИЯ

Курс «Unreal Engine: от программирования на C++ до дизайна уровней в Blueprints» предназначен для тех, кто хочет научиться создавать игры с использованием популярного игрового движка Unreal Engine. Он объединяет в себе программирование на C++ и дизайн уровней в Blueprint, чтобы обеспечить студентам полное понимание процесса создания игры на базе Unreal Engine. В рамках курса слушатели изучат основы языка программирования C++, научиться создавать собственные классы и использовать их в Unreal Engine, изучат создание компонентов и настройку логики поведения персонажей и объектов в игре. Кроме того, слушатели изучат создание уровней в Blueprint, от создания простых объектов до создания целых игровых миров, а также познакомятся с базовыми концепциями геймдизайна, чтобы создать игры, которые будут не только технически зрелыми, но и увлекательными для игроков.

В конце курса дети создадут свою первую игру на Unreal Engine, которая будет объединять в себе все изученные материалы по программированию на C++ и дизайну уровней в Blueprint. Этот курс предоставляет всем ученикам возможность получить фундаментальное понимание создания игр на Unreal Engine и достичь собственных творческих успехов. Программа направлена на формирование компетенций в соответствии с трудовыми функциями программиста.

В результате обучения выпускник программы будут иметь следующие компетенции:

- Создавать собственные игры с использованием Unreal Engine
- Разрабатывать игровые механики и функциональность на языке программирования C++
- Создавать и настраивать игровые объекты, анимации и персонажей
- Работать с инструментами дизайна уровней и создавать сложные ландшафты и архитектуру
- Настраивать освещение и звуковые эффекты, создавая уникальную атмосферу игры
- Разрабатывать и настраивать искусственный интеллект для персонажей и объектов
- Оптимизировать производительность игры и работать с плагинами для расширения функциональности Unreal Engine
- Создавать увлекательный геймплей и достигать высокого уровня игрового процесса
- Тестировать и отлаживать игровой проект перед выпуском на рынок
- Иметь все необходимые навыки для работы с Unreal Engine и создания профессиональных игровых проектов.

Рабочая программа с описанием каждого модуля

Модуль 1: Введение в Roblox и Lua (36 ч)

Модуль (описание)	Тема	Содержание	Вид учебных занятий	Объем в ак.ч.
Модуль 1: Введение в Roblox и Lua (36 ч) Модуль 1: Введение в Roblox и Lua - это модуль обучения, который предназначен для ознакомления с платформой Roblox и языком программирования Lua. В ходе этого модуля обучающиеся получат базовое представление о том, что такое Roblox, его	Тема 1.1 Введение в Unreal Engine	Введение в основы Unreal Engine . Настройка. Навигация. Размещение.	теоретические занятия	(1 ч.)
	Тема 1.2 Установка и настройка Unreal Engine	Установка и настройка Unreal Engine Установка программы. Создание проекта.	теоретические занятия	(1 ч.)
	Тема 1.3 Основы работы с редактором Unreal Engine	Основы работы с редактором Unreal Engine Навигация по интерфейсу программы. Добавление сеток в рабочую область Создание материала	теоретические занятия	(1 ч.)
	Тема 1.4 Создание простого уровня	Создание простого уровня. Создание нового уровня. Размещение актеров на уровне. Редактирование размещенных актеров.	теоретические занятия	(1 ч.)
		Задание к теме 1.4 Создание простого уровня 1. Создайте простой ландшафт, используя Landscape Tool. Разместите на нем несколько объектов, таких как деревья, камни	практические занятия	(5 ч)

возможностях и основных компонентах. Они также познакомятся с языком программирования Lua и изучат его основы, включая переменные, операторы, условные выражения, циклы и функции. Обучающиеся будут практиковаться в написании простых скриптов на Lua для управления объектами и создания интерактивности в игровых проектах на платформе Roblox. По окончании модуля		и растительность, чтобы сделать уровень более интересным. 2. Добавьте статический свет и динамический свет, чтобы уровень выглядел более реалистично. Используйте различные типы освещения, такие как точечные и направленные источники света. 3. Создайте простой скрипт, который позволяет игроку перемещаться вокруг уровня с помощью клавиатуры и мыши. Добавьте элементы взаимодействия, такие как кнопки или выключатели, чтобы сделать уровень более интересным и дать игроку больше возможностей для взаимодействия.		
		Изучить дополнительный материал по теме 1.4 Создание простого уровня https://sites.google.com/a/artdds.com/ue4ru/editor/quickstart/3	самостоятельная работа	(1 ч.)
	Тема 1.5 Работа с материалами и освещением	Работа с материалами и освещением. Общие сведения о работе с материалами (принципы, инструменты)	теоретические занятия	(1 ч.)

обучающиеся будут иметь базовые навыки программирования на Lua и понимание основ Roblox для дальнейшего развития в этой области.		Задание к теме 1.5 Работа с материалами и освещением 1. Создайте несколько материалов для объектов на уровне, используя Material Editor. Например, создайте материалы для деревьев, земли, камней и домов. Используйте различные типы текстур, такие как Normal Maps и Roughness Maps, чтобы создать реалистичный вид объектов. 2. Добавьте статический свет и динамический свет на уровне, чтобы улучшить освещение. Используйте различные типы источников света, такие как точечные и направленные, чтобы создать разные эффекты освещения. Например, можно добавить свет от солнца или лампочек. 3. Создайте простой скрипт, который позволяет игроку управлять освещением на уровне. Например, можно добавить кнопку, которая включает и выключает свет в определенных комнатах. Это поможет создать более интересный	практические занятия	(5 ч.)
---	--	---	-----------------------------	---------------

		геймплей и улучшить атмосферу на уровне.		
		Изучить дополнительный материал по теме 1.5. Работа с материалами и освещением https://www.youtube.com/watch?v=GljEAGST4eo	самостоятельная работа	(1 ч.)
	Тема 1.6 Создание первого игрового объекта	Создание первого игрового объекта. Unreal Engine 4. Создание террейна.	теоретические занятия	(1 ч.)
		Задание к теме 1.6 Создание первого игрового объекта <i>Работа со звуком и анимацией (4 ч.)</i> 1. Создайте новый уровень в Unreal Engine 4 и добавьте на него простой объект, например, куб. 2. Создайте анимацию для этого объекта, которая будет воспроизводиться при нажатии на кнопку. Например, куб может поворачиваться вокруг своей оси. 3. Добавьте звуковой эффект, который будет воспроизводиться одновременно с анимацией. Например, при повороте куба можно добавить звук вращения.	практические занятия	(2 ч.)

		Изучить дополнительный материал по теме 1.6. Создание первого игрового объекта https://www.youtube.com/watch?v=iHwo167cDDg&list=PLYpWZxEb36SgFdRsPJid4gQHjmmaGbmV	самостоятельная работа	(1 ч.)
	Тема 1.7 Работа со звуком и анимацией	Работа со звуком и анимацией. Работа со звуками, добавление шагов, фоновый шум и музыка.	теоретические занятия	(1 ч.)
		Задание к теме 1.7 Работа со звуком и анимацией 1. Скачайте несколько различных 3D моделей или текстур из сети (можно воспользоваться бесплатными ресурсами, такими как TurboSquid или CGTrader). Откройте Unreal Engine 4 и перетащите выбранные модели или текстуры в окно Content Browser. Дождитесь завершения процесса импорта. 2. Создайте сцену: Создайте новую сцену в Unreal Engine 4, выбрав пустой проект. Добавьте импортированные активы на сцену, перетащив их из Content Browser в окно Viewport. Установите активы в	практические занятия	(3 ч.)

		нужные позиции и настройте их размер и ориентацию. 3. Измените материалы: Если вы импортировали текстуры, вы можете изменить их материалы, чтобы создать различные эффекты и стили. Выберите текстуру в Content Browser и откройте Material Editor. Используйте различные узлы, такие как Texture Sample и Material Functions, чтобы создать материалы с различными эффектами, такими как блеск или шероховатость.		
		Изучить дополнительный материал по теме 1.7 Работа со звуком и анимацией https://www.youtube.com/watch?v=MI23aFs2p88	самостоятельная работа	(1 ч.)
	Тема 1.8 Импорт и использование активов	Основы импорта объектов в Unreal Engine Импорт и использование активов .	теоретические занятия	(1 ч.)
		Задание к теме 1.8 Импорт и использование активов <i>Экспорт и публикация игры (3 ч.)</i> 1. Завершите работу над игрой в Unreal Engine 4, убедившись, что она полностью готова для	практические занятия	(3 ч.)

		публикации. Убедитесь, что все активы и сцены находятся в правильных местах, все настройки игры настроены и все скрипты работают правильно. 2. Экспорт игры: Чтобы экспортировать игру, выберите File -> Package Project -> Build Configuration и выберите нужную платформу для экспорта, например, Windows или Android. Нажмите Build и дождитесь завершения процесса экспорта. В результате вы получите папку с экспортированной игрой. 3. Публикация игры: Чтобы опубликовать игру, загрузите экспортированную игру в магазине приложений. Если вы хотите опубликовать игру на Steam, например, создайте новую игру в Steamworks, загрузите экспортированные файлы игры, добавьте обложку и описание и опубликуйте игру.		
--	--	--	--	--

		Изучить дополнительный материал по теме 1.8. Импорт и использование активов https://www.youtube.com/watch?v=2naIEMCtvXA	самостоятельная работа	(1 ч.)
	Тема 1.9 Экспорт и публикация игры	Платформы для экспорта проекта. Общие настройки для экспорта проекта. Экспорт и публикация игры	теоретические занятия	(1 ч.)
		Изучить дополнительный материал по теме 1.9. Экспорт и публикация игры https://www.youtube.com/watch?v=wAo2L8QVnmY	самостоятельная работа	(1 ч.)
	Промежуточная аттестация	Практическое задание № 1. Основы Unreal Engine Шаг 1: Создайте новый проект в Unreal Engine. Шаг 2: Добавьте в проект новый актер (Actor). Шаг 3: Откройте редактор Blueprint и создайте новый Blueprint класс на основе вашего актера. Шаг 4: Откройте созданный Blueprint и добавьте в него компонент Sphere (Сфера). Шаг 5: Создайте новый скрипт (C++ класс) и свяжите его с созданным Blueprint классом. Шаг 6: Добавьте в созданный скрипт метод, который будет изменять цвет вашей сферы.	практические занятия	(3 ч.)

		Шаг 7: Сохраните изменения и закройте редактор Blueprint. Шаг 8: Перейдите в режим игры (Play Mode) и убедитесь, что при нажатии на кнопку ваша сфера меняет цвет. Шаг 9: Загрузите файл в Odin			
				Объем в ак.ч.	Объем в %
ИТОГО:			теоретические занятия	9	
			практические занятия	20	58%
			самостоятельная работа	6	17%
			промежуточная аттестация	1	
			Всего:	36	

Модуль 2: Программирование на C++ в Unreal Engine (36)

Модуль (описание)	Тема	Содержание	Вид учебных занятий	Объем в ак.ч.
Модуль 2:	Тема 2.1 Введение	Введение в программирование на C++ в Unreal Engine. Классы. Объекты.	теоретические занятия	(1 ч.)

Программирование на C++ в Unreal Engine Модуль 2: Программирование на C++ в Unreal Engine - это модуль обучения, направленный на изучение программирования на языке C++ с использованием игрового движка Unreal Engine. В ходе этого модуля обучающиеся узнают основы языка C++ и его применение в разработке игр на платформе Unreal Engine. Они изучат основные концепции языка, включая переменные, типы	в программировании на C++ в Unreal Engine	Задание к теме 2.1 <i>Введение в программирование на C++ в Unreal Engine (1 ч.)</i> 1. Создайте новый проект в Unreal Engine. 2. Используя язык программирования C++, создайте класс для игрового объекта. 3. Добавьте объект в игру и настройте его параметры, используя функции и методы класса, написанные на C++. Введение в программирование на C++ в Unreal Engine	практические занятия	(1 ч.)
	Тема 2.2 Основы языка C++	Основы языка C++. Онлайн компиляторы для C++.	теоретические занятия	(1 ч.)
		Задание к теме 2.2 Основы языка C++ <i>Основы языка C++ (2 ч.)</i> 1. Написать программу, которая запрашивает у пользователя его имя и выводит его на экран вместе с приветствием. Например, "Здравствуйте, Иван!".	практические занятия	(2 ч.)

данных, операторы, условные выражения, циклы и функции. Обучающиеся также познакомятся с архитектурой игрового движка Unreal Engine и научатся создавать собственные компоненты и функциональность с помощью языка C++. Они будут практиковаться в разработке игровых механик, обработке пользовательского ввода, управлении объектами и других аспектах программирования игр. По завершении модуля обучающиеся будут		2. Написать программу, которая находит среднее арифметическое двух целых чисел, введенных пользователем. Вывести результат на экран. 3. Написать программу, которая запрашивает у пользователя длину и ширину прямоугольника, а затем находит его площадь и периметр. Вывести результат на экран.		
	Тема 2.3 Работа с классами и объектами	Работа с классами и объектами. Глобальные классы и объекты.	теоретические занятия	(1 ч.)
		Задание к теме 2.3 Работа с классами и объектами <i>Работа с классами и объектами (2 ч.)</i> 1. Создать класс "Игрок", который содержит следующие поля: имя, количество очков, количество жизней. Написать методы, позволяющие изменять и получать значения полей, а также метод для вывода	практические занятия	(2 ч.)

обладать навыками программирования на C++ в рамках Unreal Engine и смогут создавать более сложные и функциональные игровые проекты.		информации об игроке на экран. 2. Создать класс "Оружие", который содержит следующие поля: название, урон, скорострельность. Написать методы, позволяющие изменять и получать значения полей, а также метод для вывода информации об оружии на экран. 3. Создать класс "Игровой объект", который содержит следующие поля: имя, позиция на карте, скорость, здоровье. Написать методы, позволяющие изменять и получать значения полей, а также методы для перемещения объекта, проверки его здоровья и вывода информации на экран. Создать объекты классов "Игрок" и "Оружие", а затем создать объект класса "Игровой объект", который содержит эти объекты в качестве полей.		
--	--	--	--	--

		Изучить дополнительный материал по теме 2.3 Работа с классами и объектами https://www.youtube.com/watch?v=-RHqwRQjmfU	самостоятельная работа	(1 ч.)
	Тема 2.4 Использование наследования	Использование наследования. Наследование классов.	теоретические занятия	(1 ч.)
		Задание к теме 2.4 Использование наследования <i>Использование наследования (2 ч.)</i> 1. Создать базовый класс "Персонаж", который содержит поля: имя, здоровье, скорость. Написать методы, позволяющие изменять и получать значения полей, а также метод для перемещения персонажа. 2. Создать дочерний класс "Игрок", который наследует поля и методы базового класса "Персонаж", а также содержит дополнительное поле: количество очков. Написать методы, позволяющие изменять и получать значение поля, а	практические занятия	(2 ч.)

		также метод для добавления очков игроку. 3. Создать дочерний класс "Враг", который наследует поля и методы базового класса "Персонаж", а также содержит дополнительное поле: урон. Написать методы, позволяющие изменять и получать значение поля, а также метод для атаки врагом		
		Изучить дополнительный материал по теме 2.4 Использование наследования https://www.youtube.com/watch?v=JMoF3NuMs7Q	самостоятельная работа	(1 ч.)
	Тема 2.5 Операторы и выражения в C++	Операторы и выражения в C++. Создание персонажа на C++ с нуля.	теоретические занятия	(1 ч.)
		Задание к теме 2.5 Операторы и выражения в C++ <i>Операторы и выражения в C++. (2 ч.)</i> 1. Написать программу, которая запрашивает у пользователя два числа и выводит на экран их сумму, разность, произведение	практические занятия	(2 ч.)

		и частное, используя соответствующие операторы. 2. Написать программу, которая запрашивает у пользователя радиус круга и вычисляет его площадь и длину окружности. Для вычисления использовать соответствующие математические операторы и константу Π. 3. Написать программу, которая запрашивает у пользователя два числа и проверяет, является ли первое число кратным второму. Вывести на экран соответствующее сообщение, используя оператор условия.		
		Изучить дополнительный материал по теме 2.5 Операторы и выражения в C++ https://www.youtube.com/watch?v=nzxpNL5jswQ&list=PLYpWZxEb36SjZCnTrwc-ZFCLnWg5DmvBR	самостоятельная работа	(1 ч.)
	Тема 2.6 Работа с массивами и	Работа с массивами и указателями. Указатель на строки и массивы символов.	теоретические занятия	(1 ч.)

	указателями	Задание к теме 2.6 Работа с массивами и указателями <i>Работа с массивами и указателями (3 ч.)</i> 1. Создать массив целых чисел размером 10 и заполнить его случайными значениями. Вывести на экран все элементы массива с помощью указателя. 2. Создать структуру "Студент" с полями: имя, возраст, средний балл. Создать массив студентов размером 5 и заполнить его данными с клавиатуры. Вывести на экран данные всех студентов, используя указатель на массив. 3. Создать динамический массив целых чисел размером n (значение n запрашивать у пользователя). Заполнить его случайными значениями. Вывести на экран все элементы массива, используя указатель.	практические занятия	(3 ч.)
	Тема 2.7 Использование	Использование шаблонов и контейнеров. Элементы ListView и	теоретические занятия	(1 ч.)

	шаблонов и контейнеров	GridView. Предварительные условия. Шаблон данных.		
		Задание к теме 2.7 Использование шаблонов и контейнеров <i>Использование шаблонов и контейнеров (2 ч.)</i> 1. Создать шаблонную функцию, которая принимает на вход два аргумента и возвращает результат их суммирования. Протестировать функцию на разных типах данных. 2. Создать вектор целых чисел и заполнить его случайными значениями. Используя алгоритмы STL, отсортировать вектор по возрастанию и вывести его на экран. 3. Создать класс "Счетчик", который будет считать количество созданных экземпляров этого класса. Использовать шаблон контейнера <code>std::map</code> для хранения количества	практические занятия	(2 ч.)

		экземпляров класса для каждого типа данных. Протестировать работу класса на разных типах данных.		
	Тема 2.8 Работа с файлами и потоками ввода-вывода	Работа с файлами и потоками ввода-вывода. Unreal Engine 4 пак распаковка и упаковка файлов.	теоретические занятия	(1 ч.)
		Задание к теме 2.8 Работа с файлами и потоками ввода-вывода <i>Работа с файлами и потоками ввода-вывода (2 ч.)</i> 1. Создать текстовый файл и записать в него несколько строк текста с помощью потока вывода. Затем открыть файл с помощью потока ввода и вывести на экран все строки текста. 2. Создать бинарный файл и записать в него несколько объектов пользовательского класса. Затем открыть файл с помощью потока ввода и прочитать все записанные	практические занятия	(2 ч.)

		объекты, выведя их данные на экран. 3. Написать программу, которая считывает данные из текстового файла, удаляет все гласные буквы и записывает измененный текст в новый файл. Для этого можно использовать потоки ввода-вывода и алгоритмы STL.		
		Изучить дополнительный материал по теме 2.8 Работа с файлами и потоками ввода-вывода (1 ч.) https://www.youtube.com/watch?v=0gmUtf7ekpI	самостоятельная работа	(1 ч.)
	Тема 2.9 Создание простого игрового компонента на C++	Создание простого игрового компонента на C++. Управление перемещением объекта с помощью клавиатуры или мыши.	теоретические занятия	(1 ч.)
		Задание к теме 2.9 Создание простого игрового компонента на C++ <i>Создание простого игрового компонента на C++ (2 ч.)</i>	практические занятия	(2 ч.)

		1. Создать класс игрового компонента, который будет добавлять возможность управлять перемещением объекта с помощью клавиатуры или мыши. Компонент должен содержать функции для обработки ввода и перемещения объекта в игровом мире. 2. Создать объект в игровом мире и добавить к нему созданный игровой компонент. Протестировать работу компонента, перемещая объект с помощью клавиатуры или мыши. 3. Добавить в созданный игровой компонент возможность изменять скорость перемещения объекта. Для этого можно использовать переменную класса, которая будет задавать скорость перемещения объекта, и функции для ее изменения.		
--	--	--	--	--

		Изучить дополнительный материал по теме 2.9 Создание простого игрового компонента на C++ https://www.youtube.com/watch?v=TT54UUfQu6s	самостоятельная работа	(1 ч.)
	Тема 2.10 Взаимодействие с графическим интерфейсом пользователя в Unreal Engine: создание элементов управления, обработка событий.	Изучить дополнительный материал по теме 2.10 Взаимодействие с графическим интерфейсом пользователя в Unreal Engine: создание элементов управления, обработка событий. https://www.youtube.com/watch?v=J_x7FeyDsKc	самостоятельная работа	(1 ч.)
	Промежуточная Аттестация	Практическое задание № 2. Программирование на C++ в Unreal Engine Программирование на C++ в Unreal Engine Шаг 1: Создайте новый проект в Unreal Engine. Шаг 2: Добавьте в проект новый актер (Actor).	практические занятия	(3 ч.)

		Шаг 3: Откройте редактор C++ классов и создайте новый класс на основе вашего актера. Шаг 4: Определите несколько переменных в созданном классе и проинициализируйте их. Шаг 5: Добавьте в класс метод, который будет изменять положение вашего актера. Шаг 6: Создайте объект вашего класса в игровом режиме (Play Mode) и вызовите созданный метод, чтобы переместить ваш актер в новую позицию. Шаг 7: Проверьте результат и убедитесь, что ваш актер изменил свое положение. Шаг 8: Загрузите файл в Odin			
				Объем в ак.ч.	Объем в %
ИТОГО:			теоретические занятия	9	
			практические занятия	20	55%
			самостоятельная работа	6	17%
			промежуточная аттестация	1	
			Всего:	36	

Модуль 3: Продвинутое программирование в Unreal Engine (36 ч)

Модуль (описание)	Тема	Содержание	Вид учебных занятий	Объем в ак.ч.
Модуль 3: Продвинутое программировани е в Unreal Engine Модуль 3: Продвинутое программирование в Unreal Engine - это модуль обучения, который фокусируется на более глубоком изучении программирования в рамках игрового движка Unreal Engine. В ходе этого модуля обучающиеся углубятся в продвинутые	Тема 3.1 Архитектура Unreal Engine	Архитектура Unreal Engine. Компоненты архитектуры.	теоретические занятия	(1 ч.)
		Задание к теме 3.1 Архитектура Unreal Engine 1. Изучить основные компоненты архитектуры Unreal Engine, такие как игровой мир, актеры, компоненты, контроллеры, системы и подсистемы. 2. Создать простой игровой мир, используя компоненты UE4, такие как уровни, актеры, компоненты и сетевые настройки. Разместить объекты в мире и задать им начальные параметры. 3. Реализовать простой сценарий, используя системы и подсистемы UE4, такие как физическая система, система частиц, анимации и звуковая система. Например, создать анимированную сцену, в которой объекты взаимодействуют друг с другом,	практические занятия	(2 ч.)

концепции и техники программирования, применяемые в Unreal Engine. Они изучат продвинутые возможности языка C++ в контексте разработки игр и научатся применять их для создания сложных игровых систем и функциональности. Обучающиеся будут изучать темы, такие как расширение классов игровых объектов, использование интерфейсов, работа с компонентами, оптимизация производительности		используя физические эффекты и звуковые эффекты.		
		Изучить дополнительный материал по теме 3.1 Архитектура Unreal Engine https://www.youtube.com/watch?v=qmkHkxKumj0&list=P_Latwj2blGTZ7oYmn-rhR44WHeuxId4Xec	самостоятельная работа	(1 ч.)
	Тема 3.2 Работа с игровыми объектами и компонентами	Работа с игровыми объектами и компонентами. Проекты в UE4. Добавление объектов. Сценарии.	теоретические занятия	(1 ч.)
		Задание к теме 3.2 Работа с игровыми объектами и компонентами 1. Создайте новый проект в UE4 и добавьте в него несколько игровых объектов (например, персонажей и предметов). Добавьте к ним различные компоненты (например, Mesh Component, Collision Component, Audio Component), настроив их параметры и свойства. 2. Реализуйте простой сценарий, в котором игровые объекты взаимодействуют друг с другом, используя компоненты. Например, сделайте так, чтобы персонаж мог подбирать предметы, сталкиваться с другими персонажами и взаимодействовать с окружающим миром.	практические занятия	(1 ч.)

и и другие продвинутые техники программирования. Они также будут практиковаться в разработке более сложных игровых проектов, где потребуется использование продвинутых возможностей Unreal Engine и C++. По окончании модуля обучающиеся будут обладать глубокими знаниями и навыками продвинутого программирования в Unreal Engine, что позволит им создавать высококачественны		3. Добавьте в проект новый компонент, который позволит персонажу перемещаться по игровому миру. Например, добавьте компонент Movement Component, настроив его параметры для перемещения персонажа по земле и в воздухе. Реализуйте управление персонажем с помощью клавиатуры и мыши.		
		Изучить дополнительный материал по теме 3.2 Работа с игровыми объектами и компонентами https://www.youtube.com/watch?v=CYiBxfFS-7E	самостоятельная работа	(1 ч.)
	Тема 3.3 Работа с анимацией и физикой	Работа с анимацией и физикой. Физические свойства объекта и окружающей среды. Перемещения, препятствия, столкновения.	теоретические занятия	(1 ч.)
		Задание к теме 3.3 Работа с анимацией и физикой 1. Создать игровой объект (например, персонажа) и добавить ему анимацию движения. 2. Настроить физические свойства объекта и окружающей среды, чтобы объект мог взаимодействовать с ней (например, прыгать на платформах или скользить по льду).	практические занятия	(2 ч.)

е и сложные игры с использованием этого мощного игрового движка.		3. Создать сцену, где объект может свободно перемещаться и взаимодействовать с другими объектами. Добавить несколько препятствий, на которые объект может натолкнуться или которые можно разрушить при столкновении. Включить физическую симуляцию, чтобы объекты реагировали на столкновения и силы, действующие на них.		
		Изучить дополнительный материал по теме 3.3 Работа с анимацией и физикой https://www.youtube.com/watch?v=mZhtBcKlKF0	самостоятельная работа	(1 ч.)
	Тема 3.4 Работа с интерфейсом пользователя	Работа с интерфейсом пользователя. Этапы разработки интерфейса. Проектирование.	теоретические занятия	(1 ч.)
		Изучить дополнительный материал по теме 3.4 Работа с интерфейсом пользователя https://www.youtube.com/watch?v=QFaCd4cI_cw	самостоятельная работа	(1 ч.)
	Тема 3.5 Работа с сетевыми играми	Работа с сетевыми играми. Система управления игровым процессом.	теоретические занятия	(1 ч.)
		Задание к теме 3.5 Работа с сетевыми играми 1. Разработайте простую сетевую игру для	практические занятия	(2 ч.)

		двух игроков, используя UE4. Игроки должны иметь возможность подключаться к игре через Интернет или локальную сеть. Игра может быть чем-то простым, например, двумерным аркадным шутером или гоночной игрой. 2. Разработайте систему управления игровым процессом для вашей сетевой игры. Включите в нее функционал, позволяющий игрокам перемещаться по игровому миру, взаимодействовать друг с другом, производить выстрелы или совершать другие действия в зависимости от жанра игры. 3. Разработайте систему многопользовательской игры, которая позволит игрокам общаться между собой, обмениваться сообщениями и создавать группы для совместной игры. Включите в нее функционал, позволяющий игрокам настраивать свои профили и следить за своими достижениями в игре.		
	Тема 3.6	Оптимизация производительности. Ландшафт, свет, материалы.	теоретические занятия	(1 ч.)
	Оптимизация производительности	Задание к теме 3.6	практические занятия	(3 ч.)

		Оптимизация производительности 1. Оптимизация ландшафта: создать ландшафт с использованием меньшего количества треугольников. Разбить ландшафт на отдельные компоненты, чтобы загружать только те, которые находятся ближе к игроку. 2. Оптимизация света: использовать статический свет там, где это возможно, чтобы снизить нагрузку на процессор. Динамический свет использовать только там, где это необходимо. 3. Оптимизация материалов: использовать простые материалы с меньшим количеством текстур и более простыми шейдерами. Оптимизировать настройки отображения, такие как расстояние рисования объектов, чтобы избежать ненужных расходов производительности.		
	Тема 3.7 Работа с внешними библиотеками и плагинами	Работа с внешними библиотеками и плагинами. Установка плагина. Создание приложения.	теоретические занятия	(1 ч.)
		Задание к теме 3.7 Работа с внешними библиотеками и плагинами 1. Установить и настроить плагин для работы с базой данных SQLite.	практические занятия	(3 ч.)

		2. Создать простое приложение, которое будет использовать этот плагин для сохранения и загрузки данных, связанных с игрой. 3. Протестировать приложение, убедившись в корректности сохранения и загрузки данных и отсутствии проблем с производительностью.		
	Тема 3.8 Использование асинхронных задач и потоков	Использование асинхронных задач и потоков. Написание скриптов. Добавление элементов пользовательского интерфейса.	теоретические занятия	(1 ч.)
		Задание к Теме 3.8 Использование асинхронных задач и потоков 1. Создайте сцену в UE4 с несколькими объектами. 2. Напишите скрипт, который будет перемещать объекты по сцене в асинхронном режиме, используя потоки. 3. Добавьте элементы пользовательского интерфейса, чтобы пользователь мог управлять скоростью перемещения объектов и количеством потоков, используемых для выполнения асинхронных задач.	практические занятия	(2 ч.)
		Изучить дополнительный материал по теме 3.8 Использование асинхронных задач и потоков	самостоятельная работа	(1 ч.)

		https://www.youtube.com/watch?v=fz_Sc_KcEtU		
	Тема 3.9 Разработка игры на C++ с использованием Unreal Engine	Разработка игры на C++ с использованием Unreal Engine. Настройки.	теоретические занятия	(2 ч.)
		Задание к Теме 3.9 Разработка игры на C++ с использованием Unreal Engine 1. Создайте игру в Unreal Engine, используя язык программирования C++ для разработки игровых механик. 2. Добавьте в игру элементы физики, анимации и взаимодействия с пользователем, используя функциональность Unreal Engine. 3. Оптимизируйте производительность игры, используя асинхронные задачи и потоки, а также улучшите игровой процесс, используя внешние библиотеки и плагины.	практические занятия	(2 ч.)
		Изучить дополнительный материал по теме 3.9 Р Разработка игры на C++ с использованием Unreal Engine https://www.youtube.com/watch?v=TT54UUfQu6s	самостоятельная работа	(1 ч.)
	Промежуточная аттестация	Практическое задание № 3. Продвинутое программирование в Unreal Engine Шаг 1: Создайте новый проект в Unreal Engine.	практические занятия	(3 ч.)

		Шаг 2: Откройте редактор C++ классов и создайте новый класс на основе вашего актера. Шаг 3: Определите несколько переменных в созданном классе и проинициализируйте их. Шаг 4: Добавьте в класс несколько методов, которые будут обрабатывать пользовательский ввод с клавиатуры или мыши. Шаг 5: Определите класс интерфейса пользователя (User Interface) и свяжите его с вашим классом актера. Шаг 6: Добавьте в интерфейс несколько кнопок и полей ввода для взаимодействия с пользователем. Шаг 7: Добавьте в класс актера методы, которые будут обрабатывать действия пользователя. Шаг 8: Создайте объект вашего класса в игровом режиме (Play Mode) и проверьте, что пользовательский ввод корректно обрабатывается и взаимодействие с пользователем работает правильно. Шаг 9: Добавьте в ваш класс методы для создания и управления несколькими актерами на сцене. Шаг 10: Создайте несколько объектов вашего класса и управляйте ими, используя пользовательский ввод и взаимодействие с интерфейсом пользователя. Шаг 11: Загрузите файл в Odin			
				Объем в ак.ч.	Объем в %

ИТОГО:	теоретические занятия	10	
	практические занятия	19	53%
	самостоятельная работа	6	17%
	промежуточная аттестация	1	
	Всего:	36	

Модуль 4: Дизайн уровней в Blueprints (36)

Модуль (описание)	Тема	Содержание	Вид учебных занятий	Объем в ак.ч.
Модуль 4: Дизайн уровней в Blueprints Модуль 4: Дизайн уровней в Blueprints - это модуль обучения, посвященный разработке уровней и игровых сцен с использованием	Тема 4.1 Введение в Blueprints	Введение в Blueprints. Основы BLUEPRINTS. Игровая логика.	теоретические занятия	(1 ч.)
		<i>Задание к теме 4.1</i> Введение в Blueprints 1. Создайте новый проект в Unreal Engine 4. 2. Создайте новый Blueprint-класс, представляющий игровой персонаж. Реализуйте простую игровую логику, используя Blueprints:	практические занятия	(3 ч.)

визуального программирования в Unreal Engine через Blueprints. В ходе этого модуля обучающиеся изучат основы работы с Blueprints, инструментом визуального программирования, который позволяет создавать игровые системы и поведение объектов без необходимости писать код на языке программирования. Они познакомятся с интерфейсом Blueprints, основными узлами и концепциями, а также научатся создавать уровни, размещать объекты, задавать		• Когда игрок подбирает монетку, добавьте единицу к его счётчику монет. • Когда игрок касается врага, его здоровье уменьшается на единицу. • Если здоровье игрока становится равным нулю, игра заканчивается. • Создайте несколько экземпляров монеток и врагов на игровом уровне. • Проверьте работу вашей игровой логики в редакторе Unreal Engine 4.		
		Изучить дополнительный материал по теме 4.1 Введение в Blueprints https://www.youtube.com/watch?v=wIIG9IUU46M&t=417s	самостоятельная работа	(1 ч.)
	Тема 4.2 Основы работы с Blueprints	Основы работы с Blueprints. Создание новых объектов. Создание скриптов.	теоретические занятия	(2 ч.)
		<i>Задание к теме 4.2</i> Основы работы с Blueprints 1. Создайте новый проект в Unreal Engine 4 и добавьте в него стандартную Third Person Character. 2. Используя Blueprints, создайте новый объект - Sphere. Задайте для него	практические занятия	(2 ч.)

их свойства и взаимодействие между ними. Обучающиеся также изучат продвинутые техники дизайна уровней, такие как освещение, размещение препятствий, оптимизация производительности и создание интересного геймплея. В процессе обучения они будут создавать собственные уровни, улучшая свои навыки дизайна и визуального программирования. По завершении модуля обучающиеся будут		параметры (радиус, цвет, материал и т.д.) и разместите его на сцене. 3. Создайте скрипт в Blueprints, который будет перемещать сферу вверх на определенную высоту при нажатии на кнопку клавиатуры. Сделайте так, чтобы сфера не уходила за границы экрана.		
	Тема 4.3 Создание простого игрового объекта в Blueprints	Создание простого игрового объекта в Blueprints. Добавление компонентов. Функции для управления объектами.	теоретические занятия	(1 ч.)
		<i>Задание к теме 4.3</i> Создание простого игрового объекта в Blueprints 1. Создайте новый проект в Unreal Engine 4. 2. Создайте новый Blueprint класс объекта. 3. Добавьте несколько компонентов к объекту, таких как Mesh, Collider, и любые другие необходимые компоненты для вашего объекта. 4. Создайте несколько функций в Blueprint для управления объектом. Например, функция для перемещения объекта, функция для изменения его скорости и функция для изменения его цвета.	практические занятия	(3 ч.)

способны создавать увлекательные и качественные уровни и игровые сцены с помощью Blueprints в Unreal Engine.		5. Создайте простой уровень в вашем проекте, на котором будет размещен ваш объект. 6. Разместите ваш объект на уровне и добавьте его в игру. 7. Протестируйте ваш объект, используя функции, созданные в Blueprint, и убедитесь, что он ведет себя правильно.		
		Изучить дополнительный материал по теме 4.3 Создание простого игрового объекта в Blueprints https://www.youtube.com/watch?v=iHwo167cDDg&list=PLYpWZxEb36SgFdRsPJid4gQHjmmbGmbtV	самостоятельная работа	(1 ч.)
	Тема 4.4 Использование переменных и функций в Blueprints	Использование переменных и функций в Blueprints. Управление свойствами и поведением игровых объектов.	теоретические занятия	(1 ч.)
		<i>Задание к теме 4.4</i> Использование переменных и функций в Blueprints 1. Создайте простой проект в UE4 с движком Blueprint. 2. Создайте несколько игровых объектов и задайте им различные свойства и характеристики, такие как скорость, цвет, размер и т.д.	практические занятия	(2 ч.)

		3. Используя переменные и функции в Blueprint, управляйте свойствами и поведением игровых объектов, например, при столкновении с другим объектом изменяйте цвет, при нажатии на кнопку увеличивайте скорость, и т.д.		
		Изучить дополнительный материал по теме 4.4 Использование переменных и функций в Blueprints https://www.youtube.com/watch?v=A2YzKQln7DU	самостоятельная работа	(1 ч.)
	Тема 4.5 Создание пользовательских функций и узлов	Создание пользовательских функций и узлов. Примеры пользовательских функций.	теоретические занятия	(1 ч.)
		<i>Задание к теме 4.5</i> Создание пользовательских функций и узлов 1. Создайте новый проект в Unreal Engine 4 и откройте главную сцену. 2. Создайте новый Blueprint, который будет содержать пользовательскую функцию. 3. Создайте пользовательскую функцию, которая будет принимать два входных параметра (например, "int" и "float") и возвращать их сумму. Для этого используйте узел "Add" внутри функции. 4. Создайте новый Blueprint, который будет содержать пользовательский узел.	практические занятия	(3 ч.)

		5. Создайте пользовательский узел, который будет выполнять какую-либо простую задачу (например, изменять цвет объекта в зависимости от значения входного параметра). Для этого используйте различные узлы, такие как "Set Color", "Branch" и "Make Literal Color". 6. Сохраните созданные Blueprint'ы и перейдите на главную сцену. 7. Добавьте на сцену несколько объектов и присвойте им созданные Blueprint'ы в качестве компонентов. 8. Проверьте, что созданные пользовательские функции и узлы работают правильно, и изменяют свойства объектов на сцене в соответствии с заданными параметрами.		
		Изучить дополнительный материал по теме 4.5 Создание пользовательских функций и узлов https://www.youtube.com/watch?v=szoC-H9s5xE	самостоятельная работа	(1 ч.)
	Тема 4.6 Работа со событиями и триггерами, анимацией и физикой в	Работа со событиями и триггерами, анимацией и физикой в Blueprints. Создание и настройка UI в Blueprints. Взаимодействие игрока с объектами при помощи триггеров и коллизий.	теоретические занятия	(2 ч.)
		<i>Задание к теме 4.6</i>	практические занятия	(3 ч.)

	Blueprints. Создание и настройка UI в Blueprints	Работа со событиями и триггерами, анимацией и физикой в Blueprints. Создание и настройка UI в Blueprints 1. Создать объект игрока и цель на уровне, разместить их на сетке. 2. Создать объекты для препятствий на уровне. 3. Настроить взаимодействие игрока с объектами при помощи триггеров и коллизий. Если игрок вступает в коллизию с объектом-препятствием, то он должен остановиться. 4. Создать интерфейс пользователя для ввода команд и отображения информации. На экране должно быть отображено количество сделанных ходов и кнопки для перемещения объектов на сетке. Интерфейс должен быть легко понятен и интуитивно понятен для игрока. 5. Создать анимацию для перемещения объектов на сетке. Анимация должна плавно перемещать объект от одной позиции до другой, с учетом препятствий на пути. 6. Добавить звуковые эффекты для уведомления игрока о совершенных действиях.		
--	---	---	--	--

		7. Протестировать игру на предмет ошибок и багов.		
		Изучить дополнительный материал по теме 4.6 Работа со событиями и триггерами, анимацией и физикой в Blueprints. Создание и настройка UI в Blueprints https://www.youtube.com/watch?v=1W5ozPoJkt8&list=PLatwj2blGTZ7Z_Bc40P7EjxlB9r_pWL4i	самостоятельная работа	(1 ч.)
	Тема 4.7 Разработка игры в Unreal Engine с использованием Blueprints	Разработка игры в Unreal Engine с использованием Blueprints. Обзор редактора Blueprint. Переменные в Blueprint. Вывод.	теоретические занятия	(1 ч.)
		Изучить дополнительный материал по теме 4.7 Разработка игры в Unreal Engine с использованием Blueprints https://www.youtube.com/watch?v=TaAwcm7YLq4&t=1686s	самостоятельная работа	(1 ч.)
	Консультация		практические занятия	(2 ч.)
	Промежуточная аттестация	Практическое задание № 4. Дизайн уровней в Blueprints Шаг 1: Создайте новый проект в Unreal Engine. Шаг 2: Откройте редактор уровней и создайте новый уровень.	практические занятия	(3 ч.)

		Шаг 3: Добавьте в уровень несколько статических мешей и акторов, чтобы создать окружение. Шаг 4: Откройте редактор Blueprints и создайте новый Blueprint класс на основе вашего игрового объекта (например, персонажа). Шаг 5: Определите несколько переменных в созданном классе и проинициализируйте их. Шаг 6: Добавьте в класс несколько методов, которые будут обрабатывать пользовательский ввод и управлять поведением вашего игрового объекта. Шаг 7: Свяжите ваш Blueprint класс с объектом на сцене. Шаг 8: Создайте несколько препятствий на уровне, которые игрок должен преодолеть, используя вашего персонажа. Шаг 9: Добавьте в уровень логику, которая будет определять условия прохождения уровня (например, достижение конечной точки или сбор определенного количества предметов). Шаг 10: Проверьте ваш уровень в игровом режиме (Play Mode) и убедитесь, что все логические условия работают правильно, а персонаж может пройти уровень. Шаг 11: Загрузите файл в Odin		
--	--	--	--	--

		Объем в ак.ч.	Объем в %
ИТОГО:	теоретические занятия	9	
	практические занятия	20	55%
	самостоятельная работа	6	17%
	промежуточная аттестация	1	
	Всего:	36	

Задание к теме 1.4

Создание простого уровня (5 ч)

1. Создайте простой ландшафт, используя Landscape Tool. Разместите на нем несколько объектов, таких как деревья, камни и растительность, чтобы сделать уровень более интересным.
2. Добавьте статический свет и динамический свет, чтобы уровень выглядел более реалистично. Используйте различные типы освещения, такие как точечные и направленные источники света.
3. Создайте простой скрипт, который позволяет игроку перемещаться вокруг уровня с помощью клавиатуры и мыши. Добавьте элементы взаимодействия, такие как кнопки или выключатели, чтобы сделать уровень более интересным и дать игроку больше возможностей для взаимодействия.

Задание к теме 1.5

Работа с материалами и освещением (5 ч.)

1. Создайте несколько материалов для объектов на уровне, используя Material Editor. Например, создайте материалы для деревьев, земли, камней и домов. Используйте различные типы текстур, такие как Normal Maps и Roughness Maps, чтобы создать реалистичный вид объектов.
2. Добавьте статический свет и динамический свет на уровне, чтобы улучшить освещение. Используйте различные типы источников света, такие как точечные и направленные, чтобы создать разные эффекты освещения. Например, можно добавить свет от солнца или лампочек.
3. Создайте простой скрипт, который позволяет игроку управлять освещением на уровне. Например, можно добавить кнопку, которая включает и выключает свет в определенных комнатах. Это поможет создать более интересный геймплей и улучшить атмосферу на уровне.

Задание к теме 1.6

Работа со звуком и анимацией (4 ч.)

1. Создайте новый уровень в Unreal Engine 4 и добавьте на него простой объект, например, куб.
2. Создайте анимацию для этого объекта, которая будет воспроизводиться при нажатии на кнопку. Например, куб может поворачиваться вокруг своей оси.

3. Добавьте звуковой эффект, который будет воспроизводиться одновременно с анимацией. Например, при повороте куба можно добавить звук вращения.

Задание к теме 1.7

Импорт и использование активов (3 ч.)

1. Скачайте несколько различных 3D моделей или текстур из сети (можно воспользоваться бесплатными ресурсами, такими как TurboSquid или CGTrader). Откройте Unreal Engine 4 и перетащите выбранные модели или текстуры в окно Content Browser. Дождитесь завершения процесса импорта.
2. Создайте сцену: Создайте новую сцену в Unreal Engine 4, выбрав пустой проект. Добавьте импортированные активы на сцену, перетащив их из Content Browser в окно Viewport. Установите активы в нужные позиции и настройте их размер и ориентацию.
3. Измените материалы: Если вы импортировали текстуры, вы можете изменить их материалы, чтобы создать различные эффекты и стили. Выберите текстуру в Content Browser и откройте Material Editor. Используйте различные узлы, такие как Texture Sample и Material Functions, чтобы создать материалы с различными эффектами, такими как блеск или шероховатость.

Задание к теме 1.8

Экспорт и публикация игры (3 ч.)

1. Завершите работу над игрой в Unreal Engine 4, убедившись, что она полностью готова для публикации. Убедитесь, что все активы и сцены находятся в правильных местах, все настройки игры настроены и все скрипты работают правильно.
2. Экспорт игры: Чтобы экспортировать игру, выберите File -> Package Project -> Build Configuration и выберите нужную платформу для экспорта, например, Windows или Android. Нажмите Build и дождитесь завершения процесса экспорта. В результате вы получите папку с экспортированной игрой.
3. Публикация игры: Чтобы опубликовать игру, загрузите экспортированную игру в магазине приложений. Если вы хотите опубликовать игру на Steam, например, создайте новую игру в Steamworks, загрузите экспортированные файлы игры, добавьте обложку и описание и опубликуйте игру.

Задание к теме 2.1

Введение в программирование на C++ в Unreal Engine (1 ч.)

1. Создайте новый проект в Unreal Engine.
2. Используя язык программирования C++, создайте класс для игрового объекта.

3. Добавьте объект в игру и настройте его параметры, используя функции и методы класса, написанные на C++.

Задание к теме 2.2

Основы языка C++ (2 ч.)

1. Написать программу, которая запрашивает у пользователя его имя и выводит его на экран вместе с приветствием. Например, "Здравствуйте, Иван!".
2. Написать программу, которая находит среднее арифметическое двух целых чисел, введенных пользователем. Вывести результат на экран.
3. Написать программу, которая запрашивает у пользователя длину и ширину прямоугольника, а затем находит его площадь и периметр. Вывести результат на экран.

Задание к теме 2.3

Работа с классами и объектами (2 ч.)

1. Создать класс "Игрок", который содержит следующие поля: имя, количество очков, количество жизней. Написать методы, позволяющие изменять и получать значения полей, а также метод для вывода информации об игроке на экран.
2. Создать класс "Оружие", который содержит следующие поля: название, урон, скорострельность. Написать методы, позволяющие изменять и получать значения полей, а также метод для вывода информации об оружии на экран.
3. Создать класс "Игровой объект", который содержит следующие поля: имя, позиция на карте, скорость, здоровье. Написать методы, позволяющие изменять и получать значения полей, а также методы для перемещения объекта, проверки его здоровья и вывода информации на экран. Создать объекты классов "Игрок" и "Оружие", а затем создать объект класса "Игровой объект", который содержит эти объекты в качестве полей.

Задание к теме 2.4

Использование наследования (2 ч.)

1. Создать базовый класс "Персонаж", который содержит поля: имя, здоровье, скорость. Написать методы, позволяющие изменять и получать значения полей, а также метод для перемещения персонажа.

2. Создать дочерний класс "Игрок", который наследует поля и методы базового класса "Персонаж", а также содержит дополнительное поле: количество очков. Написать методы, позволяющие изменять и получать значение поля, а также метод для добавления очков игроку.
3. Создать дочерний класс "Враг", который наследует поля и методы базового класса "Персонаж", а также содержит дополнительное поле: урон. Написать методы, позволяющие изменять и получать значение поля, а также метод для атаки врагом.

Задание к теме 2.5

Операторы и выражения в C++. (2 ч.)

1. Написать программу, которая запрашивает у пользователя два числа и выводит на экран их сумму, разность, произведение и частное, используя соответствующие операторы.
2. Написать программу, которая запрашивает у пользователя радиус круга и вычисляет его площадь и длину окружности. Для вычисления использовать соответствующие математические операторы и константу Пи.
3. Написать программу, которая запрашивает у пользователя два числа и проверяет, является ли первое число кратным второму. Вывести на экран соответствующее сообщение, используя оператор условия.

Задание к теме 2.6

Работа с массивами и указателями (3 ч.)

1. Создать массив целых чисел размером 10 и заполнить его случайными значениями. Вывести на экран все элементы массива с помощью указателя.
2. Создать структуру "Студент" с полями: имя, возраст, средний балл. Создать массив студентов размером 5 и заполнить его данными с клавиатуры. Вывести на экран данные всех студентов, используя указатель на массив.
3. Создать динамический массив целых чисел размером n (значение n запрашивать у пользователя). Заполнить его случайными значениями. Вывести на экран все элементы массива, используя указатель.

Задание к теме 2.7

Использование шаблонов и контейнеров (2 ч.)

1. Создать шаблонную функцию, которая принимает на вход два аргумента и возвращает результат их суммирования. Протестировать функцию на разных типах данных.

2. Создать вектор целых чисел и заполнить его случайными значениями. Используя алгоритмы STL, отсортировать вектор по возрастанию и вывести его на экран.
3. Создать класс "Счетчик", который будет считать количество созданных экземпляров этого класса. Использовать шаблон контейнера `std::map` для хранения количества экземпляров класса для каждого типа данных. Протестировать работу класса на разных типах данных.

Задание к теме 2.8

Работа с файлами и потоками ввода-вывода (2 ч.)

1. Создать текстовый файл и записать в него несколько строк текста с помощью потока вывода. Затем открыть файл с помощью потока ввода и вывести на экран все строки текста.
2. Создать бинарный файл и записать в него несколько объектов пользовательского класса. Затем открыть файл с помощью потока ввода и прочитать все записанные объекты, выведя их данные на экран.
3. Написать программу, которая считывает данные из текстового файла, удаляет все гласные буквы и записывает измененный текст в новый файл. Для этого можно использовать потоки ввода-вывода и алгоритмы STL.

Задание к теме 2.9

Создание простого игрового компонента на C++ (2 ч.)

1. Создать класс игрового компонента, который будет добавлять возможность управлять перемещением объекта с помощью клавиатуры или мыши. Компонент должен содержать функции для обработки ввода и перемещения объекта в игровом мире.
2. Создать объект в игровом мире и добавить к нему созданный игровой компонент. Протестировать работу компонента, перемещая объект с помощью клавиатуры или мыши.
3. Добавить в созданный игровой компонент возможность изменять скорость перемещения объекта. Для этого можно использовать переменную класса, которая будет задавать скорость перемещения объекта, и функции для ее изменения.

Задание к теме 2.10

Взаимодействие с графическим интерфейсом пользователя в Unreal Engine: создание элементов управления, обработка событий. (2 ч.)

1. Создать графический интерфейс пользователя с использованием элементов управления, таких как кнопки, текстовые поля и выпадающие списки. Разместить элементы на экране и настроить их внешний вид и поведение.
2. Добавить обработчики событий для элементов управления, чтобы реагировать на действия пользователя. Например, при нажатии на кнопку можно запустить определенное действие или изменить значение параметра в игровом мире.
3. Протестировать работу графического интерфейса пользователя, взаимодействуя с элементами управления и проверяя правильность работы обработчиков событий. При необходимости настроить внешний вид и поведение элементов управления.

Задание к теме 3.1

Архитектура Unreal Engine. (2 ч.)

1. Изучить основные компоненты архитектуры Unreal Engine, такие как игровой мир, актеры, компоненты, контроллеры, системы и подсистемы.
2. Создать простой игровой мир, используя компоненты UE4, такие как уровни, актеры, компоненты и сетевые настройки. Разместить объекты в мире и задать им начальные параметры.
4. Реализовать простой сценарий, используя системы и подсистемы UE4, такие как физическая система, система частиц, анимации и звуковая система. Например, создать анимированную сцену, в которой объекты взаимодействуют друг с другом, используя физические эффекты и звуковые эффекты.

Задание к теме 3.2

Работа с игровыми объектами и компонентами (1 ч.)

1. Создайте новый проект в UE4 и добавьте в него несколько игровых объектов (например, персонажей и предметов). Добавьте к ним различные компоненты (например, Mesh Component, Collision Component, Audio Component), настроив их параметры и свойства.
2. Реализуйте простой сценарий, в котором игровые объекты взаимодействуют друг с другом, используя компоненты. Например, сделайте так, чтобы персонаж мог подбирать предметы, сталкиваться с другими персонажами и взаимодействовать с окружающим миром.

3. Добавьте в проект новый компонент, который позволит персонажу перемещаться по игровому миру. Например, добавьте компонент Movement Component, настроив его параметры для перемещения персонажа по земле и в воздухе. Реализуйте управление персонажем с помощью клавиатуры и мыши.

Задание к теме 3.3

Работа с анимацией и физикой (2 ч.)

1. Создать игровой объект (например, персонажа) и добавить ему анимацию движения.
2. Настроить физические свойства объекта и окружающей среды, чтобы объект мог взаимодействовать с ней (например, прыгать на платформах или скользить по льду).
3. Создать сцену, где объект может свободно перемещаться и взаимодействовать с другими объектами. Добавить несколько препятствий, на которые объект может натолкнуться или которые можно разрушить при столкновении. Включить физическую симуляцию, чтобы объекты реагировали на столкновения и силы, действующие на них.

Задание к теме 3.5

Работа с сетевыми играми (2 ч.)

1. Разработайте простую сетевую игру для двух игроков, используя UE4. Игроки должны иметь возможность подключаться к игре через Интернет или локальную сеть. Игра может быть чем-то простым, например, двумерным аркадным шутером или гоночной игрой.
2. Разработайте систему управления игровым процессом для вашей сетевой игры. Включите в нее функционал, позволяющий игрокам перемещаться по игровому миру, взаимодействовать друг с другом, производить выстрелы или совершать другие действия в зависимости от жанра игры.
3. Разработайте систему многопользовательской игры, которая позволит игрокам общаться между собой, обмениваться сообщениями и создавать группы для совместной игры. Включите в нее функционал, позволяющий игрокам настраивать свои профили и следить за своими достижениями в игре.

Задание к теме 3.6

Оптимизация производительности (3 ч.)

1. Оптимизация ландшафта: создать ландшафт с использованием меньшего количества треугольников. Разбить ландшафт на отдельные компоненты, чтобы загружать только те, которые находятся ближе к игроку.
2. Оптимизация света: использовать статический свет там, где это возможно, чтобы снизить нагрузку на процессор. Динамический свет

использовать только там, где это необходимо.

3. Оптимизация материалов: использовать простые материалы с меньшим количеством текстур и более простыми шейдерами. Оптимизировать настройки отображения, такие как расстояние рисования объектов, чтобы избежать ненужных расходов производительности.

Задание к теме 3.7

Работа с внешними библиотеками и плагинами (3 ч.)

1. Установить и настроить плагин для работы с базой данных SQLite.
2. Создать простое приложение, которое будет использовать этот плагин для сохранения и загрузки данных, связанных с игрой.
3. Протестировать приложение, убедившись в корректности сохранения и загрузки данных и отсутствии проблем с производительностью.

Задание к теме 3.8

Использование асинхронных задач и потоков (2 ч.)

1. Создайте сцену в UE4 с несколькими объектами.
2. Напишите скрипт, который будет перемещать объекты по сцене в асинхронном режиме, используя потоки.
3. Добавьте элементы пользовательского интерфейса, чтобы пользователь мог управлять скоростью перемещения объектов и количеством потоков, используемых для выполнения асинхронных задач.

Задание к теме 3.9

Разработка игры на C++ с использованием Unreal Engine (2 ч.)

1. Создайте игру в Unreal Engine, используя язык программирования C++ для разработки игровых механик.
2. Добавьте в игру элементы физики, анимации и взаимодействия с пользователем, используя функциональность Unreal Engine.
3. Оптимизируйте производительность игры, используя асинхронные задачи и потоки, а также улучшите игровой процесс, используя внешние библиотеки и плагины.

Задание к теме 4.1

Введение в Blueprints (2 ч.)

1. Создайте новый проект в Unreal Engine 4.
2. Создайте новый Blueprint-класс, представляющий игровой персонаж.

Реализуйте простую игровую логику, используя Blueprints:

- Когда игрок подбирает монетку, добавьте единицу к его счётчику монет.
- Когда игрок касается врага, его здоровье уменьшается на единицу.
- Если здоровье игрока становится равным нулю, игра заканчивается.
- Создайте несколько экземпляров монеток и врагов на игровом уровне.
- Проверьте работу вашей игровой логики в редакторе Unreal Engine 4.

Задание к теме 4.2

Основы работы с Blueprints (2 ч.)

1. Создайте новый проект в Unreal Engine 4 и добавьте в него стандартную Third Person Character.
2. Используя Blueprints, создайте новый объект - Sphere. Задайте для него параметры (радиус, цвет, материал и т.д.) и разместите его на сцене.
3. Создайте скрипт в Blueprints, который будет перемещать сферу вверх на определенную высоту при нажатии на кнопку клавиатуры. Сделайте так, чтобы сфера не уходила за границы экрана.

Задание к теме 4.3

Создание простого игрового объекта в Blueprints (3 ч.)

1. Создайте новый проект в Unreal Engine 4.
2. Создайте новый Blueprint класс объекта.
3. Добавьте несколько компонентов к объекту, таких как Mesh, Collider, и любые другие необходимые компоненты для вашего объекта.
4. Создайте несколько функций в Blueprint для управления объектом. Например, функция для перемещения объекта, функция для изменения его скорости и функция для изменения его цвета.
5. Создайте простой уровень в вашем проекте, на котором будет размещен ваш объект.
6. Разместите ваш объект на уровне и добавьте его в игру.
7. Протестируйте ваш объект, используя функции, созданные в Blueprint, и убедитесь, что он ведет себя правильно.

Задание к теме 4.4

Использование переменных и функций в Blueprints (2 ч.)

1. Создайте простой проект в UE4 с движком Blueprint.
2. Создайте несколько игровых объектов и задайте им различные свойства и характеристики, такие как скорость, цвет, размер и т.д.
3. Используя переменные и функции в Blueprint, управляйте свойствами и поведением игровых объектов, например, при столкновении с другим объектом изменяйте цвет, при нажатии на кнопку увеличивайте скорость, и т.д.

Задание к теме 4.5

Создание пользовательских функций и узлов (3 ч.)

1. Создайте новый проект в Unreal Engine 4 и откройте главную сцену.
2. Создайте новый Blueprint, который будет содержать пользовательскую функцию.
3. Создайте пользовательскую функцию, которая будет принимать два входных параметра (например, "int" и "float") и возвращать их сумму. Для этого используйте узел "Add" внутри функции.
4. Создайте новый Blueprint, который будет содержать пользовательский узел.
5. Создайте пользовательский узел, который будет выполнять какую-либо простую задачу (например, изменять цвет объекта в зависимости от значения входного параметра). Для этого используйте различные узлы, такие как "Set Color", "Branch" и "Make Literal Color".
6. Сохраните созданные Blueprint'ы и перейдите на главную сцену.
7. Добавьте на сцену несколько объектов и присвойте им созданные Blueprint'ы в качестве компонентов.
8. Проверьте, что созданные пользовательские функции и узлы работают правильно, и изменяют свойства объектов на сцене в соответствии с заданными параметрами.

Задание к теме 4.6

Работа со событиями и триггерами, анимацией и физикой в Blueprints. Создание и настройка UI в Blueprints (3 ч.)

1. Создать объект игрока и цель на уровне, разместить их на сетке.
2. Создать объекты для препятствий на уровне.

3. Настроить взаимодействие игрока с объектами при помощи триггеров и коллизий. Если игрок вступает в коллизию с объектом-препятствием, то он должен остановиться.
4. Создать интерфейс пользователя для ввода команд и отображения информации. На экране должно быть отображено количество сделанных ходов и кнопки для перемещения объектов на сетке. Интерфейс должен быть легко понятен и интуитивно понятен для игрока.
5. Создать анимацию для перемещения объектов на сетке. Анимация должна плавно перемещать объект от одной позиции до другой, с учетом препятствий на пути.
6. Добавить звуковые эффекты для уведомления игрока о совершенных действиях.
7. Протестировать игру на предмет ошибок и багов.

Календарный план-график реализации образовательной программы

Первый модуль с 01.10.2023 - 30.11.2023

Второй модуль: с 01.12.2023 - 31.01.2024

Третий модуль: с 01.02.2024 - 31.03.2024

Четвертый модуль: с 01.04.2024 - 31.05.2024

Занятия проводятся 2-3 раза в неделю по 2 академических часа, длительность одного академического часа – 45 минут.

1 занятие 2 академических часа

Календарно-тематическое планирование

<i>№</i>	<i>Тема и № модуля</i>	<i>Тема занятия</i>	<i>Кол-во занятий*</i>	<i>Кол-во часов</i>	<i>Дата</i>
1	Модуль 1:	Тема 1.1 Введение в Unreal Engine	1	1	2.10.23

Основы Unreal

2	Engine 01.10.2023 - 30.11.2023	Тема 1.2 Установка и настройка Unreal Engine	1	1	2.10.23
3		Тема 1.3 Основы работы с редактором Unreal Engine	1	1	5.10.23
4		Тема 1.4 Создание простого уровня	3	6	5.10.23 9.10.23 12.10.23 16.10.23
5		Тема 1.5 Работа с материалами и освещением	3	6	16.10.23 19.10.23 23.10.23 26.10.23
6		Тема 1.6 Создание первого игрового объекта	2	3	26.10.23 27.10.23
7		Тема 1.7 Работа со звуком и анимацией	2	4	8.11.23 13.11.23
8		Тема 1.8 Импорт и использование активов	2	4	16.11.23 20.11.23
9		Тема 1.9 Экспорт и публикация игры	2	4	23.11.23 27.11.23
10		Промежуточная аттестация	1	1	29.11.23

11	Модуль 2: Программирование на C++ в Unreal Engine 01.12 2023 - 31.01.2024	Тема 2.1 Введение в программирование на C++ в Unreal Engine	1	2	1.12.23
12		Тема 2.2 Основы языка C++	2	3	4.12.23 7.12.23
13		Тема 2.3 Работа с классами и объектами	2	3	7.12.23 11.12.23
14		Тема 2.4 Использование наследования	2	3	14.12.23 18.12.23
15		Тема 2.5 Операторы и выражения в C++	2	3	18.12.23 21.12.23
16		Тема 2.6 Работа с массивами и указателями	2	4	25.12.23 27.12.23
17		Тема 2.7 Использование шаблонов и контейнеров	2	3	9.01.24 12.01.24
18		Тема 2.8 Работа с файлами и потоками ввода-вывода	2	3	12.01.24 15.01.24
19		Тема 2.9 Создание простого игрового компонента на C++	2	3	18.01.24 22.01.24
20		Тема 2.10 Взаимодействие с графическим интерфейсом пользователя в Unreal Engine: создание элементов управления, обработка событий.	2	3	22.01.24 25.01.24

21		Промежуточная аттестация	1	1	29.01.24
22	Модуль 3: Продвинутое программиров ание в Unreal Engine 01.02.2024 - 31.03.2024	Тема 3.1 Архитектура Unreal Engine	2	3	1.02.24 5.02.24
23		Тема 3.2 Работа с игровыми объектами и компонентами	2	3	5.02.24 8.02.24
24		Тема 3.3 Работа с анимацией и физикой	2	3	12.02.24 15.02.24
25		Тема 3.4 Работа с интерфейсом пользователя	2	3	15.02.24 19.02.24
26		Тема 3.5 Работа с сетевыми играми	2	3	22.02.24 26.02.24
27		Тема 3.6 Оптимизация производительности	2	4	26.02.24 29.02.24 4.03.24
28		Тема 3.7 Работа с внешними библиотеками и плагинами	2	4	4.03.24 7.03.24 11.03.24
29		Тема 3.8 Использование асинхронных задач и потоков	2	3	11.03.24 14.03.24

30		Тема 3.9 Разработка игры на C++ с использованием Unreal Engine	2	4	18.03.24 21.03.24
31		Промежуточная аттестация	1	1	23.01.24
32	Модуль 4: Дизайн уровней в Blueprints 01.02.2024 - 31.03.2024	Тема 4.1 Введение в Blueprints	2	4	1.04.24 4.04.24
33		Тема 4.2 Основы работы с Blueprints	2	4	8.04.24 11.04.24
34		Тема 4.3 Создание простого игрового объекта в Blueprints	2	4	15.04.24 18.04.24
35		Тема 4.4 Использование переменных и функций в Blueprints	2	3	22.04.24 25.04.24
36		Тема 4.5 Создание пользовательских функций и узлов	2	4	25.04.24 29.04.24 4.05.24
37		Тема 4.6 Работа со событиями и триггерами, анимацией и физикой в Blueprints. Создание и настройка UI в Blueprints	3	5	4.05.24 6.05.24 8.05.24
38		Тема 4.7 Разработка игры в Unreal Engine с использованием Blueprints	2	4	13.05.24 16.05.24

39		Консультация	1	2	<i>20.05.24</i>
40		Промежуточная аттестация	1	1	<i>23.05.24</i>

Учебно-методические материалы

<i>Наименование поля</i>	<i>Допустимые значения полей</i>	<i>Значение полей</i>	<i>Значение полей</i>	<i>Значение полей</i>	<i>Значение полей</i>
<i>Порядковый номер модуля</i>	<i>строка не менее 10 символов</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>
<i>Методы, формы и технологии</i>	<i>строка не менее 10 символов</i>	Программа реализуется в формате смешанного обучения. Синхронные занятия включают интерактивные лекции и практические занятия, предусматривающие групповую и индивидуальную работу слушателей. Для организации асинхронной работы слушателей используются записи синхронных занятий, презентации, конспекты лекций, практические задания и тестовые вопросы, размещаемые в электронном курсе на платформе электронного обучения LMS Odin.			
<i>Методические разработки</i>	<i>строка не менее 10 символов</i>	Для изучения программы используется электронный курс в LMS Odin.			
<i>Материалы модуля</i>	<i>строка не менее 10 символов</i>	Для изучения программы используется электронный курс в LMS Odin. Электронный курс включает информационно-организационные материалы по программе (учебно-тематический план, календарный график работы по программе, информацию о преподавателях программы, чат для объявлений и вопросов, сведения о результатах обучения), набор видеолекций, презентации к лекциям, набор ссылок на внешние ресурсы, систему заданий с инструкциями, списки основной и дополнительной литературы.			

<i>Учебная литература</i>	<i>строка не менее 10 символов</i>	Литература: 1. К. Виллет. Unreal Engine. Профессиональная разработка игр. 2016. 2. Х. Дикман. Разработка игр в Unreal Engine 4. 2017. 3. А. Адамович, А. Сергиенко. Unreal Engine 4: игровая разработка в примерах. 2018. 4. Б. Баранов, Ю. Зуев. Unreal Engine 4: программирование и разработка игр. 2018. 5. Д. Шейд. Unreal Engine 4 для профессионалов. 2016. 6. Д. Рид. Разработка компьютерных игр в Unreal Engine 4. 2017. 7. М. Миткевич. Unreal Engine 4 для начинающих. 2018. 8. Ш. Паттерсон. Unreal Engine 4 для дизайнеров уровней. 2017. 9. А. Рязанов. Unreal Engine 4: разработка приложений виртуальной реальности. 2019. 10. М. Пирогов. Unreal Engine 4 для мобильных устройств. 2019
---------------------------	------------------------------------	---

Материально-технические условия реализации программы

<i>Наименование поля</i>	<i>Допустимые значения полей</i>	<i>Значение полей</i>	<i>Значение полей</i>	<i>Значение полей</i>	<i>Значение полей</i>
<i>Порядковый номер модуля</i>	<i>строка не менее 10 символов</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>
<i>Наименование требуемого оборудования</i>	<i>строка не менее 2 символов</i>	Оборудование для синхронных занятий: персональный компьютер (рекомендуется) / мобильный телефон / планшет; наушники, микрофон и камера (обязательно). Для работы на платформе электронного обучения LMS Odin рекомендуется использовать персональный компьютер. Для успешной реализации Программы необходимо, чтобы рабочее место обучающегося и преподавателя включали в себя: • компьютеры, обеспечивающие возможность работы с мультимедийным контентом: воспроизведение видеоизображений, качественный стереозвук в наушниках, речевой ввод с микрофона и др. периферийное оборудование: • принтер (черно/белой печати, формата А4); • устройства для ввода визуальной информации (сканер, цифровой фотоаппарат, web-камера и пр.);			

		• устройства создания графической информации (графический планшет), используемые для создания и редактирования графических объектов, ввода рукописного текста; • акустические колонки; • оборудование, обеспечивающее подключение к сети Интернет (комплект оборудования для подключения к сети Интернет, сервер). Сеть: скорость соединения от 2 Мб/с.
Наименование требуемого программного обеспечения	строка не менее 2 символов	Программное обеспечение компьютера: • операционная система семейства MacOS или Windows; • программа Python, • файловый менеджер (в составе операционной системы или др.); • почтовый клиент (в составе операционных систем или др.); • браузер (в составе операционных систем или др.); • мультимедиа проигрыватель (в составе операционной системы или др.); • антивирусная программа; • программа-архиватор; • программа-переводчик; • программа интерактивного общения; • текстовый редактор; • растровый графический редактор; • звуковой редактор; • редактор Web-страниц.
Электронные информационные ресурсы	строка не менее 10 символов	

<i>Электронные образовательные ресурсы</i>	<i>строка не менее 10 символов</i>	
--	------------------------------------	--

Адреса и координаты

№ п/п	Название адрес	Адрес	Код адреса	Долгота	Широта
целое число	строка	строка	Целое число	вещественное число	вещественное число
1	Школа №	ул.			