

**УПРАВЛЕНИЕ ОБРАЗОВАНИЯ МЭРИИ ГОРОДА ЧЕРЕПОВЦА**

организация дополнительного образования детей и взрослых  
«КОДИНГЦЕНТР»



УТВЕРЖДЕНО  
ИП Горошкова В.А.  
Приказ № 1-П  
от «31» августа 2023г.

Дополнительная общеобразовательная общеразвивающая  
программа технической направленности  
**PROFI.LEVEL**

**Направленность:** техническая  
**Уровень программы:** базовый  
**Возраст обучающихся:** 15-17 лет  
**Срок реализации программы:** 1 уч.год  
**Разработчики:**  
педагог дополнительного образования  
Табунов П.А.

Программа утверждена на заседании педагогического совета  
организации дополнительного образования детей и взрослых  
«КОДИНГЦЕНТР» «31» августа 2023г.

Вологодская область  
город Череповец  
2023/2024 уч. год

## Оглавление

РАЗДЕЛ 1 «КОМПЛЕКС ОСНОВНЫХ ХАРАКТЕРИСТИК ОБРАЗОВАНИЯ: ОБЪЕМ, СОДЕРЖАНИЕ, ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ». ....	3
РАЗДЕЛ № 2 «КОМПЛЕКС ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИХ УСЛОВИЙ, ВКЛЮЧАЮЩИЙ ФОРМЫ АТТЕСТАЦИИ» .....	16
УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ .....	23

## **РАЗДЕЛ 1 «КОМПЛЕКС ОСНОВНЫХ ХАРАКТЕРИСТИК ОБРАЗОВАНИЯ: ОБЪЕМ, СОДЕРЖАНИЕ, ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ».**

Образовательная программа «PROFI.LEVEL» реализуется в рамках курса, реализующегося организации дополнительного образования детей и взрослых «КОДИНГЦЕНТР».

### **Программа разработана в соответствии с требованиями:**

- Федеральный закон Российской Федерации от 29 декабря 2012 г. № 273-ФЗ «Об образовании в Российской Федерации».
- Стратегия развития воспитания в Российской Федерации на период до 2025 года, утвержденная распоряжением Правительства Российской Федерации от 29 мая 2015 г. N 996-р.
- Проект Концепции развития дополнительного образования детей до 2030 года.
- Приоритетный проект «Доступное дополнительное образование для детей», утвержденный 30 ноября 2016 года протоколом заседания президиума при Президенте РФ.
- Федеральный проект «Успех каждого ребенка», утвержденный 7 декабря 2018 г.
- Приказ Министерства просвещения Российской Федерации от 09 ноября 2018г. №196 «Об утверждении порядка организации и осуществления образовательной деятельности по дополнительным. общеобразовательным программам».
- Приказ Министерства образования и науки РФ от 9 января 2014 г. № 2 «Об утверждении порядка применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ».
- Постановление Главного государственного санитарного врача Российской Федерации от 28 сентября 2020 года N 28 «Об утверждении

санитарных правил СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи».

— Письмо Минобрнауки России от 28.04.2017 N ВК-1232/09 «О направлении методических рекомендаций» (вместе с «Методическими рекомендациями по организации независимой оценки качества дополнительного образования детей»).

— Методические рекомендации по проектированию дополнительных общеразвивающих программ от 18.11.2015 г. Министерство образования и науки РФ.

**Направление образовательной программы:** техническое.

Одна из задач – содействовать воспитанию нового поколения, отвечающего по своему уровню развития и образу жизни условиям информационного общества. Для этого учащимся необходимо анализировать информацию, выявлять в ней факты и проблемы, самостоятельно ставить задачи, структурировать и преобразовывать информацию, использовать ее для решения учебных и жизненных задач.

**Актуальность данной программы.** В рамках школьного курса программирование на языках высокого уровня изучается обзорно и только в старших классах. За отведенное количество часов невозможно сформировать стойкие навыки решения задач с помощью одной из систем программирования. Специфика обучения детей в старшей школе и далее в вузовском образовании предполагает владение обучающимися всеми возможными технологиями решения информационных задач с помощью компьютера. А потому для успешности наших выпускников в обучении необходимо предоставить им возможность освоить технологию программирования на одном из языков высокого уровня.

**Педагогическая целесообразность.** Формирование и развитие творческих способностей, удовлетворение индивидуальных потребностей в интеллектуальном, нравственном и физическом совершенствовании,

формирование культуры здорового и безопасного образа жизни. Программа предполагает работу над индивидуальными и коллективными работами. Каждый учащийся любого уровня подготовки и способностей в процессе обучения чувствует себя важным звеном общей цепи (системы), от которого зависит исполнение коллективной работы в целом. Доля ответственности каждого учащегося в этом процессе очень значима. Учащийся, осознавая эту значимость, старается исполнить свою часть работы достойно, что способствует формированию чувства ответственности и значимости каждого участника школьного коллектива.

Учащийся сам ответственен за выполнение всех поставленных творческих задач, и в его руках находится решение целого процесса создания дизайн-проекта (например, создание своего сайта).

Таким образом, педагогическая целесообразность образовательной программы заключается в формировании у учащегося чувства ответственности в исполнении своей индивидуальной функции в коллективном процессе, с одной стороны, и формировании самодостаточного проявления своих творческих способностей в работе с использованием всех изученных технологий при выполнении индивидуальных заданий.

Содержание курса сочетает в себе три существующих сейчас основных подхода в обучении информатики в школе:

- **алгоритмический** (программистский), связанный с развитием мышления школьников;
- **«пользовательский»**, связанный с формированием компьютерной грамотности, информационной культуры, подготовкой учащихся к практической деятельности в условиях широкого использования информационных технологий;
- **кибернетический**, связанный с формированием мировоззренческих представлений о роли информации в управлении.

Основа курса – личностная, практическая и продуктивная направленность занятий. Одна из целей обучения – предоставить ученикам

возможность личностного самоопределения и самореализации по отношению к стремительно развивающимся информационным технологиям и ресурсам.

### **Цели программы:**

- Познакомить учащихся с ролью программного обеспечения и его видами.
- Сформировать целостное представление об организации данных для эффективной алгоритмической обработки.
- Развить логическое мышление.
- Реализация математических способностей учащихся в ходе составления программ на языке программирования.

### **Задачи курса:**

#### **Обучающие:**

- Познакомить учащихся с основными алгоритмическими конструкциями и правилами их записи, с основными способами организации данных.
- Научить учащихся составлять и записывать алгоритмы с использованием соответствующих алгоритмических конструкций.
- Научить распознавать необходимость применения той или иной алгоритмической конструкции при решении задач.
- Научить организовывать данные для эффективной алгоритмической обработки.
- Научить учащихся разрабатывать алгоритмы и реализовывать их на языке программирования C# и C++.
- Научить создавать простые приложения в среде разработки Unity 3D, Unreal Engine.
- Научить учащихся осуществлять отладку и тестирование программы.

#### **Развивающие:**

- Формировать новый тип мышления – операционный, который направлен на выбор оптимальных решений;
- Предоставление возможности узнать новое в области компьютерного программирования;

— Формирование представления о роли компьютерного программирования в развитии общества, изменении содержания и характера деятельности человека.

**Воспитательные:**

- Повышение общекультурного уровня учащихся;
- Вооружение учащихся правильным методологическим подходом к познавательной и практической деятельности;
- Выделение и раскрытие роли информационных технологий и компьютеров в развитии современного общества;
- Привитие навыков сознательного и рационального использования компьютера в своей учебной, а затем и профессиональной деятельности;
- Формирование эмоционально-ценностного отношения к миру, к себе;
- Воспитание у учащихся стремления к овладению техникой исследования;
- Воспитание трудолюбия, инициативности и настойчивости в преодолении трудностей.

**Возраст обучающихся:** 15-17 лет.

**Срок реализации.** Программа рассчитана на 1 учебный год. Общий объем часов – 72ч.

**Формы занятий:**

- групповые – для всей группы, посвященные практическим и теоретическим занятиям;
- индивидуальные консультации в рамках групповых занятий;
- дистанционные – для всей группы и/или индивидуально в установленное время.

Программа может корректироваться с учетом имеющейся материально-технической базы и контингента обучающихся. Количество обучающихся в группе до 10 человек.

**Режим занятий.** Обучающиеся занимаются 1 раз в неделю по 2 академических часа. Во время перерывов, предусмотренных нормами СанПин

2.4.3648-20, учащиеся выполняют комплекс «Упражнений для снятия усталости глаз», также проводятся снимающие мышечную усталость физкультминутки.

Количество учащихся в учебной группе до 10 человек.

На основе учебной программы преподаватель организует всю работу по предмету.

Основной формой обучения является практическая работа. Структура занятий в объединении для обучающихся с ограниченными возможностями здоровья составлена так, что 20-25% времени идёт на теорию и 75-80%- на практическую работу.

В основе, адаптированной дополнительной общеобразовательной общеразвивающей программы следующие принципы:

- учет индивидуальных особенностей и возможностей учащихся с ОВЗ;
- систематичность и последовательность, от простого к сложному;
- интеграция образовательных областей;
- уход от жесткого регламентирования обучения;
- создание развивающей предметной среды;
- использование многообразных форм организации обучения;
- обеспечение взаимосвязи занятий с повседневной жизнью детей, их самостоятельной деятельностью (игровой, художественной, конструктивной и др.);
- широкое использование методов, активизирующих мышление, воображение, поисковую деятельность;
- создание эмоционально значимых для детей ситуаций;
- выделение в качестве ведущей диалоговой формы общения педагога и ребенка, детей между собой, что формирует уважение, доверие и совместное творчество.

**Психолого-педагогические особенности детей с ОВЗ, умственной отсталостью, физические особенности детей.**



Дети-инвалиды и дети с ограниченными возможностями здоровья традиционно рассматриваются как одна из наиболее уязвимых категорий детей с точки зрения их социальной успешности и личностного развития.

Нарушения психической деятельности при олигофрении составляют основу дефекта. Патология психического статуса у этих детей связана с недоразвитием всей познавательной деятельности и особенно мышления. Процесс восприятия этого мира малоактивен. Выработанные условные связи у них непрочны и быстро угасают. Обычно детали, второстепенные признаки предметов и явлений плохо или совсем не фиксируются.

Перечисленные выше нарушения восприятия и представлений отрицательно сказываются на формировании высших психических функций, которые составляют основу интеллектуальной деятельности. Детям бывает трудно найти сходство или различие между предметами по характерным признакам, высказать суждение о предмете, явлении, создавшейся ситуации, самостоятельно оценить ситуацию и найти в ней свое место. Это нередко является причиной неадекватных реакций и недостаточно осмысленных форм поведения.

Познавательная деятельность умственно отсталых учащихся состоит в недоразвитии словесно-логического мышления.

Отмечаются также нарушения волевой сферы. Внешне это выражается в хаотическом поведении, преобладании движений и действий произвольного характера. Кроме того, у них нередко наблюдается стойкое проявление негативизма. Слабость воли у многих детей выражается еще и в том, что они легко подчиняются чужому влиянию (как правило, отрицательному), не проявляют настойчивости и инициативы в достижении цели. Их поведение импульсивно.

Произвольное же внимание у большинства из них отличается неустойчивостью – дети легко отвлекаются на любой посторонний раздражитель, что очень осложняет их обучение. Но в процессе обучения и воспитания внимание становится более устойчивым, расширяется его объем.

Этому весьма способствует овладение навыками самообслуживания и выполнения трудовых заданий.

К нарушениям психической деятельности нужно отнести и наблюдающиеся у умственно отсталого расстройства эмоциональной сферы, которая имеет большое значение в процессе познания, поскольку активизирует работу мышления. Чувства таких детей малодифференцированы, бедны.

Одно из типичных нарушений даже при легкой степени умственной отсталости – это расстройство речи. Слабо развита мелкая моторика.

Адаптированная общеобразовательная общеразвивающая программа технической направленности PROFI.LEVEL раскрывает и развивает творческие способности ребёнка с ограниченными возможностями здоровья. Занятия по данной программе развивают внимание, концентрацию, художественный вкус, фантазию, трудолюбие.

Все виды работы, представленные в программе, развивают у обучающихся способность мыслить, мыслить алгоритмически, способность работать под управлением сознания.

У детей развивается чувство коллективизма, ответственности за свой труд, уважение к труду других. Особенно важно, что дети познают значимость своего труда, его полезность для окружающих. Очень важно совершенствовать эстетический вкус детей, развивать чувство прекрасного, поддерживать творческое начало в деятельности каждого ребенка.

**Ожидаемые результаты и способы определения их результативности:**

По окончании курса обучения, обучающиеся будут знать:

- правила техники безопасности при проведении работ с компьютерной техникой и периферийным оборудованием;
- основы языка программирования C#, C++, основы работы в среде Unity 3D, Unreal Engine;

Будут уметь:

- применять полученные знания при работе с компьютерной техникой и периферийным оборудованием;
- применять полученные знания в своей деятельности;
- работать с прикладным ПО;
- простые программы на языке C#, C++;
- разрабатывать мини-проекты, мини-приложения в среде разработки Unity 3D, Unreal Engine;

**Формы подведения итогов реализации программы:**

Контроль знаний осуществляется после каждого информационного блока в виде итогового задания, позволяющего оценить знания обучающихся, полученных в результате освоения блока. Допустимые формы контроля – тестирование, практическое задание, экзамен, защита проекта.

## УЧЕБНЫЙ ПЛАН

**Возраст обучаемых:** дети 15-17 лет

**Длительность обучения:** 1 уч.год

**Форма обучения:** очно-дистанционная

**Режим занятий:** 1 раза в неделю (по 2 академических часа)

**Недельная нагрузка:** 2 часа в неделю

№	Название раздела	Всего	Т	П	Форма аттестации/контроля
1	Вводное занятие	1	1		-
3	Раздел 1. Unity. Углубленное изучение среды с использованием языка программирования C#	26	8	18	Творческий проект по итогам двух модулей
4	Раздел 2. Основы разработки моделей в среде 3D-моделирования Blender	12	2	10	
3	Раздел 3. Unreal Engine 4. Углубленное изучение среды с использованием языка программирования C++	26	8	18	Творческий проект по итогам двух модулей
4	Раздел 4. Blender. Продвинутое моделирование.	13	2	11	

### Программы учебных модулей

#### Раздел 1. Unity. Углубленное изучение среды с использованием языка программирования C#

1. Знакомство. Первая программа «Hello World».
2. Переменные, типы данных int, float.
3. Основные арифметические операции с типами данных
4. Ветвление.

5. Циклы. Циклы for и while.
6. Массивы.
7. Тип данных string.
8. Функции.
9. Введение в Unity
10. Программирование движения объектов
11. Программная генерация мира
12. Управление объектами
13. Обработка столкновений (триггеров)
14. Создание игрового процесса
15. Создание магазина в игре
16. Пользовательский интерфейс
17. Основные элементы пользовательского интерфейса
18. Создание UI (пользовательского интерфейса)
19. Эффекты (системы частиц, следы)

### **Раздел 3. Unreal Engine. Углубленное изучение среды с использованием языка программирования C++**

1. Установка движка
2. Импорт ассетов
3. Создание материалов
4. Использование Blueprints для создания объектов с простейшими функциями
5. Настройка камеры с видом сверху
6. Создание управляемого игроком актора с простыми движениями
7. Настройка ввода игрока
8. Создание элемента, исчезающего при контакте с игроком
9. Управление текстурами, яркость и цвет
10. Использование экземпляров материалов для быстрого создания вариаций

11. Использование динамических экземпляров материалов
12. Создание HUD-дисплея, на котором отображается счётчик и таймер
13. Отображение HUD на экране
14. Бесконечное перемещение игрока вперёд
15. Генерация препятствий
16. Рандомизация препятствия для создания вариаций
17. Импорт меша со скелетом
18. Импорт анимации
19. Создание Animation Blueprint
20. Воспроизведение музыки
21. Воспроизведение звуковых эффектов
22. Управление с помощью UI громкостью музыки и звуковых эффектов
23. Создание системы частиц
24. Создание ИИ-сущности
25. Создание и использование деревьев поведения и blackboard
26. Использование AI Perception
27. Создание поведения, чтобы Pawn мог ходить и атаковать врагов
28. Создание Pawn с видом от первого лица, который сможет двигаться и осматриваться вокруг
29. Создание оружия
30. Нанесение урона акторам

## **Раздел 2 и 4. Blender. Начальный и продвинутый уровень. 1**

### **Тема 1. Введение, основы работы, основные инструменты и модификаторы**

Камера - перспектива и ортография. Системы координат. 3Д курсор. Магнит. Удаление. Объединение и разделение. Режим редактирования для нескольких объектов. Выдавливание (extrude). Вдавливание (inset). Фаска. Вращение (spin tool). Скольжение. Петля. Изменение масштаба и его применение. Модификаторы (Subdivision surface, Bevel и т.д.)

## **Тема 2. Основы моделирования**

Правила построения полигональной сетки. Основные проблемы, и как с ними бороться. Оптимизация отображения модели во viewport. Создание иерархической структуры объектов проекта. Использование примитивов для создания сложных форм. Коррекция геометрии после применения модификатора boolean. Подготовка модели к текстурированию. Основы анимации.

## **Тема 3. Работа с материалами и текстурами**

Изучение основных нодов. Создание многослойных материалов. Создание масок для смешивания различных слоев. Использование готовых текстур для ускорения. Создание (рисование) текстур от руки непосредственно на модели.

## **РАЗДЕЛ № 2 «КОМПЛЕКС ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИХ УСЛОВИЙ, ВКЛЮЧАЮЩИЙ ФОРМЫ АТТЕСТАЦИИ»**

### **Условия реализации программы**

Реализация программы производится в полном объеме, качество обучения соответствует установленным требованиям, применяемые формы, средства, методы обучения и воспитания соответствует возрастным, психофизическим особенностям, склонностям, способностям, интересам и потребностям обучающихся.

Основными формами обучения являются теоретические, практические занятия.

Теоретические занятия проводятся в оборудованных учебных кабинетах, оснащенных соответствующим оборудованием, техническими средствами обучения.

Используется 1 учебный кабинет с достаточной пропускной способностью, в соответствии с установленными для них требованиями безопасности. Оснащённых компьютерной техникой, интерактивным оборудованием, имеется выход в Интернет.

На всех компьютерах установлено базовое лицензионное программное обеспечение включающее операционную систему, пакет прикладных программ и антивирусное программное обеспечение, а также часть бесплатно распространяемых программ (графические редакторы, программы моделирования). Наполняемость учебной группы не превышает 10 человек

### **Контрольно-оценочные средства**

#### **Перечень вопросов к итоговому зачёту**

**Раздел 1. Unity. Углубленное изучение среды с использованием языка программирования C#**

1. Что такое Unity и для чего оно используется?



2. Как создать новый проект в Unity?
3. Какие основные компоненты входят в объект в Unity?
4. Что такое префабы (Prefabs) и для чего они используются?
5. Каким образом можно перемещать объекты в Unity?
6. Что такое компонент Rigidbody и как он используется?
7. Каким образом можно задать поворот объекта в Unity?
8. Каким образом можно изменить размер объекта в Unity?
9. Что такое коллайдеры (Colliders) и зачем они нужны?
10. Как создать анимацию объекта в Unity?
11. Как добавить звуковые эффекты в Unity?
12. Как реализовать взаимодействие между объектами в Unity?
13. Каким образом можно добавить эффект частиц в Unity?
14. Что такое скрипты (Scripts) и для чего они используются в Unity?
15. Как создать новый скрипт в Unity и привязать его к объекту?
16. Как осуществить передвижение объекта с помощью скрипта?
17. Каким образом можно обращаться к компонентам объекта в скрипте?
18. Как реализовать управление игроком с помощью клавиатуры или других устройств ввода?
19. Каким образом можно обнаруживать столкновения между объектами в Unity с помощью скриптов?
20. Каким образом можно загружать и использовать ресурсы, такие как текстуры и модели, в Unity?

### **Раздел 3. Unreal Engine. Углубленное изучение среды с использованием языка программирования C++**

1. Что такое Unreal Engine и для чего он используется?
2. Какие основные компоненты составляют архитектуру Unreal Engine?
3. Какой язык программирования используется для разработки на Unreal Engine?
4. Что такое классы и объекты в Unreal Engine?

5. Что такое Blueprints в Unreal Engine и какой язык программирования используется в них?
6. Как создать новый проект в Unreal Engine с использованием C++?
7. Каким образом Unreal Engine обрабатывает события и что такое Event Driven Programming?
8. В чем разница между Actor и Pawn в Unreal Engine?
9. Какие компоненты используются для создания пользовательского интерфейса в Unreal Engine?
10. Что такое компоненты (Components) в Unreal Engine и как их использовать в C++?
11. Каким образом Unreal Engine работает с физикой и коллизиями?
12. Как добавить анимацию и контролировать ее в Unreal Engine с использованием C++?
13. Каким образом Unreal Engine обрабатывает искусственный интеллект (AI)?
14. Как в Unreal Engine реализуется система ввода (Input)?
15. Как управлять звуком и аудиоэффектами в Unreal Engine?
16. Как в Unreal Engine рассчитывается и управляется источник света (Lighting)?
17. Что такое Blueprint Function Library и как его использовать в C++?
18. Как создать собственный компонент (Custom Component) в Unreal Engine?
19. Как подключить сторонние библиотеки или плагины в Unreal Engine с использованием C++?
20. Как создать собственный класс в Unreal Engine с использованием C++ и как использовать его в проекте?

## **Раздел 2 и 4. Blender. Начальный и продвинутый уровень. 1**

1. Что такое Blender и для чего он используется?

2. Что такое 3D-моделирование? Какое 3D-моделирование можно делать в Blender?
3. Как добавить новый объект в сцену в Blender?
4. Какими способами можно выбрать объекты в Blender?
5. Что такое UV-развертка и как ее можно сделать в Blender?
6. Какие оси существуют в Blender и как их можно изменить?
7. Как управлять и двигать камерой в Blender?
8. Что такое анимация в Blender и как ее создавать?
9. Как добавить новый материал к объекту в Blender?
10. Что такое освещение в Blender и как его настраивать?
11. Как добавить физическую симуляцию в Blender?
12. Как создать дым или огонь в Blender?
13. Что такое рендеринг в Blender и как его настроить?
14. Как сделать прозрачный фон при рендеринге в Blender?
15. Какими способами можно создать текст в Blender?
16. Какими средствами можно изменить форму объекта в Blender?
17. Как использовать модификаторы в Blender?
18. Какими способами можно создавать частицы в Blender?
19. Что такое арматура (skeleton) в Blender и как ее использовать?
20. Как экспортировать готовую сцену или объект из Blender в другой формат файлов?

### **Перечень вопросов для самостоятельного изучения**

#### **Раздел 1. Освоение языка программирования C# в игровом движке**

##### **Unity**

1. Операции. Преобразование типов
2. Массивы параметров и params
3. Область видимости переменных
4. Перечисления, классы и объекты, структуры
5. Модификаторы доступа

6. Перегрузка методов
7. Статические члены класса, константы и поля для чтения
8. Перегрузка операторов
9. Наследование
10. Преобразование типов
11. Анонимные типы
12. Асинхронные методы, `async`, `await`
13. Шейдеры и визуальные эффекты: изучение создания пользовательских шейдеров для достижения потрясающих визуальных эффектов и придания играм уникального стиля.
14. Искусственный интеллект: изучение создания алгоритмов и поведения ИИ для противников и неписей в играх.
15. Физическая симуляция: изучение возможностей физического моделирования и создание реалистичных физических эффектов в играх.
16. Сетевая игра: изучение создания многопользовательских игр, работы через сеть и синхронизации действий игроков.
17. Оптимизация производительности: изучение методов оптимизации кода, ресурсов и процессов, чтобы достичь высокой производительности игр.
18. Расширение функциональности Unity: изучение расширения возможностей Unity с помощью плагинов и пользовательских инструментов.
19. Создание пользовательского интерфейса: изучение создания эффективного и интуитивно-понятного пользовательского интерфейса для игр.
20. Аудио в играх: изучение использования звуковых эффектов и музыки для создания качественного звукового дизайна в играх.
21. VR и AR разработка: изучение создания виртуальной и дополненной реальности при помощи Unity, используя специальные SDK.
22. Продвинутые спецэффекты: изучение создания сложных и впечатляющих спецэффектов, таких как пламя, вода, дым и другие.

### **Раздел 3. Unreal Engine. Углубленное изучение среды с использованием языка программирования C++**

1. Работа с физикой и частицами в Unreal Engine: изучите возможности создания реалистичных эффектов, используя физические симуляции и системы частиц.
2. Расширение функциональности Unreal Engine с помощью C++: изучите программирование на C++ в контексте Unreal Engine для создания собственных классов и расширения функционала движка.
3. Создание современных игровых механик: изучите продвинутые принципы дизайна игровых механик, таких как стрельба на бегу, разрушаемость окружения, сложные системы ИИ.
4. Создание динамической и интерактивной окружающей среды: изучите создание динамической окружающей среды, используя системы взаимодействия с предметами, динамическое освещение и эффекты.
5. Создание мультиплеерных игр: изучите создание мультиплеерных игр с помощью различных сетевых режимов Unreal Engine и работы с распределенными объектами.
6. Оптимизация производительности Unreal Engine: изучите различные методы оптимизации производительности игр, включая управление ресурсами, локализацию, оптимизацию трассировки лучей и т.д.
7. Создание процедурной генерации контента: изучите методы генерации процедурного контента, такие как создание случайных уровней, текстур, анимаций и моделей.
8. Создание VR-приложений и игр: изучите создание виртуальной реальности (VR) приложений и игр с использованием Unreal Engine, включая работу с контроллерами и механиками взаимодействия.
9. Разработка AR-приложений и игр: изучите создание дополненной реальности (AR) приложений и игр, используя функции распознавания маркеров, отображение объектов в реальном времени и другие AR-технологии.

10. Создание качественных кат-сцен и визуализации: изучите техники создания кинематографических кат-сцен и высококачественной визуализации в Unreal Engine, включая освещение, материалы, пост-обработку, эффекты и камеру.

## **Раздел 2 и 4. Blender. Начальный и продвинутый уровень. 1**

1. Основы интерфейса Blender
2. Моделирование 3D-объектов в Blender
3. Создание и настройка материалов и текстур
4. Использование света и освещения в Blender
5. Анимация объектов и персонажей в Blender
6. Работа с ключевыми кадрами и временной шкалой
7. Создание реалистичных сцен и окружений
8. Работа с камерами и создание видео в Blender
9. Рендеринг и настройка параметров рендера в Blender
10. Использование модификаторов для изменения формы объектов
11. Создание физических эффектов, таких как жидкость и частицы
12. Создание анимации ходьбы и движения персонажей
13. Работа с UV-разверткой и текстурным маппингом
14. Создание волос и волосковых систем
15. Импорт и экспорт моделей в различных форматах
16. Использование блендформов для деформации объектов
17. Создание интерактивных элементов и управления сценой
18. Создание игрового контента и взаимодействия в Blender Game Engine
19. Использование Python для автоматизации задач и создания собственных плагинов в Blender
20. Работа с архитектурными проектами и создание виртуальных туров.

**УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ**

1. Гибсон Бонд Гибсон Бонд Джереми. Unity и C#. Геймдев от идеи до реализации / Гибсон Бонд Гибсон Бонд Джереми. – Питер. – 2019. – 928с.
2. Мюллер Джон Поль, Семпф. Билл. C# для чайников. / Мюллер Джон Поль, Семпф. Билл. - Диалектика-Вильямс – 2019. – 608с.
3. Прайс Марк . C# 7 и .NET Core. Кросс-платформенная разработка для профессионалов / Прайс Марк. – Питер. – 2018. – 640с.
4. Хокинг Джозеф. Unity в действии. Мультиплатформенная разработка на C# / Хокинг Джозеф. – Питер.
5. Blender для начинающих, курс– [Электронный ресурс] URL: <https://gitjournal.tech/uroki-blender-3d-dlja-nachinajushhih-na-russkom-jazyke/>
6. Куксон А. Разработка игр на Unreal Engine 4 за 24 часа – Москва : Эксмо, 2019. – 528 с.
7. Макеффри М. Unreal Engine VR для разработчиков – Москва : Эксмо, 2019. – 256 с.
8. Сантелло С. – Разработка RPG в Unreal v4.7 – Packt Publishing Ltd. – 2016. – 425 с.
9. Шериф У. Изучаем C++ создавая игры в UE4 - Packt Publishing – 2015

ДОКУМЕНТ ПОДПИСАН  
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

СВЕДЕНИЯ О СЕРТИФИКАТЕ ЭП

Сертификат 726242342903868691666490759959119263676517201275

Владелец Горошкова Виктория Александровна

Действителен с 16.09.2023 по 15.09.2024