


МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
ФГБОУ ВО «Башкирский государственный университет»
Институт непрерывного образования

УТВЕРЖДАЮ
Проректор по молодежной политике
и международной деятельности
Т.Б. Великжанина
(подпись)
М.П.
« 04 » 2022 г.

СОГЛАСОВАНО
Проректор по научной
и инновационной работе


(подпись) С.А. Мустафина.
(инициалы, фамилия)

« 18 » 04 2022 г.

СОГЛАСОВАНО
И.о. директора института
непрерывного образования


(подпись) Е.П. Кислова

« 18 » 04 2022 г.

**ПРОГРАММА
ПРОФЕССИОНАЛЬНОЙ ПЕРЕПОДГОТОВКИ**

Программирование и дизайн игр и приложений виртуальной реальности

(наименование программы)

Программист

(наименование присваиваемой квалификации (в соответствии с профстандартом))

УФА 2022 год

1. ОБЩАЯ ХАРАКТЕРИСТИКА ПРОГРАММЫ

1.1. Цель реализации программы

Программа имеет целью формирование профессиональных компетенций в соответствии с трудовыми функциями программиста.

Программа разработана в соответствии с профессиональным стандартом «Программист», утвержденным приказом Министерства труда и социальной защиты Российской Федерации от 18 ноября 2013 г. № 679н (зарегистрирован Министерством юстиции Российской Федерации 18 декабря 2013 г., регистрационный № 30635), с изменениями, внесенными приказом Министерства труда и социальной защиты Российской Федерации от 12 декабря 2016 г. № 727н (зарегистрирован Министерством юстиции Российской Федерации 13 января 2017 г., регистрационный № 45230).

1.2. Характеристика нового вида профессиональной деятельности, новой квалификации

1.2.1. Область профессиональной деятельности: разработка программного обеспечения.

Об Связь, информационные и коммуникационные технологии.

1.2.2. Объекты профессиональной деятельности выпускников: *системы 3D моделирования и программирования, системы разработки компьютерных игр для шлемов виртуальной реальности, компьютерные системы с поддержкой технологии виртуальной реальности.*

1.2.3. Типы задач профессиональной деятельности выпускника: *проектная.*

1.2.4. Задачи профессиональной деятельности выпускника:

- работа с Blender – программным обеспечением для создания трёхмерной компьютерной графики,
- создание элементов графического пользовательского интерфейса,
- создание 3D-моделей, анимации и сцен для приложений, игр, фильмов, иллюстраций,
- написание программного кода с использованием языка программирования C# и игрового движка Unity,
- разработка VR приложений с использованием игрового движка Unity в соответствии с техническим заданием,
- проверка и отладка программного кода,
- рефакторинг и оптимизация программного кода,
- работа в составе команды над разработкой игр и приложений VR.

Присваивается квалификация: Программист

1.3. Требование к результатам освоения программы

В результате освоения программы слушатель должен приобрести следующие профессиональные компетенции:

ПК-1: Применяет стандарты и методики при оформлении программного кода

Знать: внутреннюю организацию программного кода, правила и рекомендации оформления программного кода, стандарты и соглашения о написании программного кода на C#.

Уметь: использовать стандарты и соглашения при написании программного кода, оптимально и эффективно организовывать программный код.

Владеть: навыками оформления программного кода в соответствии с принятыми правилами и стандартами.

ПК-2: Оценивает возможности применения виртуальной и дополненной реальности

Знать: историю появления и эволюцию технологии виртуальной реальности (VR), области применения технологии VR, программные и технические средства, используемые при работе с виртуальной реальностью.

Уметь: использовать программные и технические средства для работы с виртуальной реальностью; разрабатывать приложения и игры, использующие виртуальную реальность.

Владеть: представлениями о технологиях виртуальной и дополненной реальности, инструментами и SDK для разработки приложений виртуальной реальности.

ПК-3: Применяет языки программирования

Знать: синтаксис языка программирования C#, особенности программирования на этом языке, пакеты программ в области программирования (Microsoft Visual Studio Community, Unity); общие принципы и навыки практического применения объектно-ориентированного программирования на C#.

Уметь: проектировать и разрабатывать 3D приложения и игры для ПК; разрабатывать игры и приложения виртуальной реальности на основе SteamVR Plugin с использованием игрового движка Unity и языка программирования C#, проводить их отладку и тестирование; разрабатывать интерфейс пользователя для игр и приложений; проводить рефакторинг и оптимизацию программного кода; использовать возможности имеющейся технической и/или программной архитектуры.

Владеть: методами использования в профессиональной деятельности языка программирования C#; инструментами игрового движка Unity; современными методами практического программирования; умениями и навыками использования библиотек объектов (классов) игрового движка Unity для решения практических задач.

ПК-4: Применяет интегрированные среды разработки (IDE)

Знать: основы языка программирования C# и основы работы с IDE (Microsoft Visual Studio).

Уметь: использовать одну или несколько современных IDE для написания программного кода.

Владеть: навыками использования IDE для разработки приложений виртуальной реальности в соответствии с поставленной задачей.

ПК-5: Использует 3д моделирование

Знать: интерфейс программы Blender; основы полигонального моделирования; процесс создания развертки и текстуринга моделей.

Уметь: работать с программой 3D моделирования Blender; создавать элементы графического пользовательского интерфейса; создавать 3D-модели, анимации и сцены для приложений и игр, выполнять рендеринг модели или сцены.

Владеть: инструментами среды моделирования Blender, навыками использования модификаторов, материалов и света.

1.4. Требования к уровню подготовки поступающего на обучение, необходимому для освоения программы

Лица, имеющие высшее или среднее профессиональное образование, получающие высшее или среднее профессиональное образование. Программа рассчитана на обучение слушателей без наличия базовых знаний, умений и навыков в области работы с программным обеспечением для создания трёхмерной компьютерной графики, но желательно иметь базовые знания в области программирования.

1.5. Трудоемкость обучения

Трудоемкость обучения по данной программе – 256 часов, включая все виды аудиторной и внеаудиторной (самостоятельной) учебной работы слушателя и 80 контактных часов.

1.6. Форма обучения

Форма обучения очно-заочная, с применением дистанционных образовательных технологий и электронного обучения.

2. СОДЕРЖАНИЕ ПРОГРАММЫ

2.1. Учебный план

Дисциплины	Общая трудоемкость, час	По учебному плану с использованием дистанционных образовательных технологий, час.								СРС, час	Промежуточная (при наличии) и итоговая аттестация
		Аудиторные занятия, час.*				Дистанционные занятия, час.					
		всего	из них			всего	из них				
			ЛК	ПЗ	ЛР		ЛК	ПЗ	ЛР		
1	2	3	4	5	6	7	8	9	10	11	12
3D моделирование	72	24	10	14		42	18	24		6	Зачет
Анимация персонажей	36	14	6	8		18	8	10		4	Зачет
Разработка 3D игр и приложений на Unity	72	14	6	8		52	12	40		6	Зачет
Проектирование и архитектура приложений VR	72	24	6	18		42	12	30		6	Экзамен
Итоговая аттестация	4	4		4							Защита проектной работы
<i>Итого</i>	256	80	28	52		154	50	104		22	

2.2. Учебно-тематический план

№ п/п	Тема и содержание (краткая аннотация занятия)	Форма изучения материалов (лекции, семинарские занятия, лабораторные работы, самостоятельная работа)	Кол-во часов	Формируемая компетенция (из профессионального стандарта)
1	2	3	4	5
	3D моделирование		72	
1.	Интерфейс Blender	лекции, практические занятия, самостоятельная работа	4	ПК-5
2.	Материалы и свет	лекции, практические занятия, самостоятельная работа	6 8	ПК-5
3.	Драфт	лекции, практические занятия, самостоятельная работа	6 6 2	ПК-5
4.	Полигональное моделирование	лекции, практические занятия, самостоятельная работа	6 8 2	ПК-5
5.	Модификаторы	лекции, практические занятия, самостоятельная работа	6 8	ПК-5
6.	Рендеринг	лекции, практические занятия, самостоятельная работа	8 2	ПК-5

	Анимация персонажей		36	
7.	Скелетная анимация	лекции, практические занятия, самостоятельная работа	6 6	ПК-5
8.	Лицевая анимация	лекции, практические занятия, самостоятельная работа	4 6 2	ПК-5
9.	Камера, свет, рендеринг	лекции, практические занятия, самостоятельная работа	4 6 2	ПК-5
	Разработка 3D игр и приложений на Unity		72	
10.	Игровой движок Unity. Возможности на сегодняшний день	лекции, практические занятия, самостоятельная работа	2	ПК-3
11.	Архитектура проекта. Структура проекта в Unity. Работа со сценой	лекции, практические занятия, самостоятельная работа	2	ПК-3
12.	Процесс разработки игрового проекта в среде Unity	лекции, практические занятия, самостоятельная работа	8 36 4	ПК-1, ПК-3, ПК-4
13.	Создание элементарного интерфейса с использованием Canvas	лекции, практические занятия, самостоятельная работа	6 12 2	ПК-1, ПК-3, ПК-4
	Проектирование и архитектура приложений VR		72	
14.	Виртуальная реальность (VR). Оборудование для работы VR	лекции, практические занятия, самостоятельная работа	2	ПК-2
15.	Способы работы с VR	лекции, практические занятия, самостоятельная работа	2	ПК-2
16.	Создание проекта VR на Unity. Взаимодействие с объектами при работе с VR	лекции, практические занятия, самостоятельная работа	8 36 4	ПК-1, ПК-2, ПК-3, ПК-4
17.	Создание интерфейсов пользователя при работе с VR	лекции, практические занятия, самостоятельная работа	6 12 2	ПК-1, ПК-2, ПК-3, ПК-4

2.3 Дисциплинарное содержание программы

№	Дисциплина, раздел программы	Связь с результатами обучения
1	3D моделирование	ПК-5
2	Анимация персонажей	ПК-5
3	Разработка 3D игр и приложений на Unity	ПК-1, ПК-3, ПК-4
4	Проектирование и архитектура приложений VR	ПК-1, ПК-2, ПК-3, ПК-4

3. РАБОЧИЕ ПРОГРАММЫ ДИСЦИПЛИН

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ «3D МОДЕЛИРОВАНИЕ»

1. **Цель дисциплины:** знакомство с основами моделирования в программе Blender, формирование необходимых умений и навыков для создания 3D моделей объектов и их оптимизации.

2. **Планируемые результаты обучения, соотнесенные с формируемыми компетенциями:**

Компетенция	Планируемые результаты обучения
ПК-5: Использует 3д моделирование	Знать: интерфейс программы Blender; основы полигонального моделирования; процесс создания развертки и текстуринга моделей. Уметь: работать с программой 3D моделирования Blender; создавать элементы графического пользовательского интерфейса; создавать 3D-модели, анимации и сцены для приложений и игр, выполнять рендеринг модели или сцены. Владеть навыками: использования инструментов среды моделирования Blender, использования модификаторов, материалов и света.

3. Трудоемкость дисциплины

Общая трудоемкость дисциплины составляет 72 ч. По дисциплине предусмотрены лекции в объеме 28 часов, практические занятия – 38 часов, СРС – 6 ч. и ИКСР слушателя – 2 ч на слушателя. Форма итогового контроля: зачет.

4. Содержание дисциплины

№, Наименование темы	Содержание лекций (количество часов)	Наименование лабораторных работ (количество часов)	Наименование практических занятий или семинаров (количество часов)	Виды СРС (количество часов)
1	2	3	4	5
Интерфейс Blender	4	0	0	0
Материалы и свет	6	0	8	0
Драфт	6	0	6	2
Полигональное моделирование	6	0	8	2
Модификаторы	6	0	8	0
Рендеринг	0	0	8	2
Всего 72	28	0	38	6

5. Учебно-методическое обеспечение СРС

Согласно действующему Учебному плану на самостоятельную работу слушателя по дисциплине «3D моделирование» отводится 6 часов. СРС по данной дисциплине предполагает подготовку к текущему и итоговому контролю, выполнение практических заданий.

Самостоятельная работа предусматривает изучение отдельных разделов и тем учебной дисциплины, изучение программного обеспечения, подготовку к зачету, выполнение заданий.

Текущая и опережающая СРС:

- работа с лекционным материалом, поиск и анализ литературы и электронных источников информации по отдельным темам;
- подготовка к практическим занятиям, выполнение домашних заданий;
- изучение тем, вынесенных на самостоятельную проработку;
- выполнение практических заданий;
- подготовка к итоговой аттестации.

К каждой теме приведены контрольные задания. Для самостоятельной подготовки предложен теоретический материал, непосредственно связанный с выполняемыми практическими работами, а также список рекомендуемой литературы и источников в сети интернет.

Форма контроля текущей и опережающей СРС – устный опрос, проверка выполнения практических заданий, задание-кейс.

6. Организационно-педагогические условия реализации дисциплины:

а) Материально-технические условия

Режим реализации: очно-заочная, с применением дистанционных образовательных технологий и электронного обучения.

Минимальные требования к компьютеру:

- Intel Core i3 3240 (или аналог от AMD) и выше.
- Оперативная память DDR3 и выше объемом 8 Гб и более.
- Видеокарта с объемом памяти от 2 Гб и выше (для разработки приложений допускается использовать встроенную видеокарту).
- Наличие от 50Гб свободного места на SSD или HDD.
- Монитор (или экран ноутбука) с разрешением экрана 1440 x 900 точек и глубиной цвета 32 bit (рекомендуемое разрешение экрана 1920 x 1080).
- Акустическая система или наушники.
- Доступ в Интернет со скоростью от 10 Мбит/с и выше.

Информационные технологии:

- Мультимедиа-технологии.
- Интернет-технологии (наличие выхода в интернет с целью поиска современной практико-ориентированной и учебной литературы, а также учебных видеокурсов по дисциплине).
- Дистанционные технологии обучения на базе СДО.

Программное обеспечение:

- Операционная система Windows 10.
- Интернет-браузер Яндекс.
- Программа для 3D моделирования Blender.
- OBS Studio – бесплатная программа с открытым исходным кодом для записи видео и потокового вещания.

- Discord – кроссплатформенная проприетарная система мгновенного обмена сообщениями с поддержкой VoIP и видеоконференций.
- Система дистанционного обучения Moodle.

б) Учебно-методическое и информационное обеспечение

Основная литература:

1. Хэсс Ф. Практическое пособие Blender 3.0 для любителей и профессионалов. Моделинг, анимация, VFX, видеомонтаж. – М.: Солон-Пресс. 2022. – 300 с.
2. Слаква А. Инструменты моделирования в Blender. [Электронный ресурс] – URL: <https://disk.yandex.ru/d/p8Sm9Eods24y0Q>

Дополнительная литература:

1. Blender Secrets Vol. 1 + 2 + 3. [Электронный ресурс] – URL: <https://online-courses.club/blender-secrets-vol-1-2-3-ebook/>

Перечень ресурсов сети Интернет:

1. Blender 3.1 Reference Manual. [Электронный ресурс] – URL: <https://docs.blender.org/manual/en/latest/index.html>
2. Официальный сайт Blender. – URL: <https://www.blender.org/>
3. Бесплатные текстуры Poly Haven. [Электронный ресурс] – URL: <https://polyhaven.com/textures>
4. Бесплатные текстуры ambientCG. [Электронный ресурс] – URL: <https://ambientcg.com/list>

7. Методические указания для обучающихся

К каждой теме дисциплины приведены упражнения для закрепления материала. Для самостоятельной подготовки предложен теоретический материал, непосредственно связанный с выполняемыми работами, а также список рекомендуемой литературы и источников в сети интернет.

Передача необходимых теоретических знаний и формирование основных представлений по каждой теме дисциплины происходит с использованием лекционных занятий. Лекционный материал закрепляется в ходе практических занятий. Изучение дисциплины сопровождается самостоятельной работой слушателей в форме изучения пройденного лекционного материала, подготовки к практическим занятиям и выполнения индивидуальных практических заданий в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков обучающихся.

Если самостоятельно не удастся разобраться в материале, необходимо сформулировать вопрос и задать преподавателю на занятии.

Необходимо уделить внимание следующим понятиям: драфт, вьюпорт, полигон, режим редактирования, объектный режим, UV-развертка, рендеринг, модификаторы.

При реализации различных видов учебной работы также используются:

- Электронные лекции в СДО Moodle.
- Электронные образовательные ресурсы (как предлагаемые интернет, так и авторские).

Практические задания являются одним из видов самостоятельной работы слушателей. Они выполняются после прослушивания лекций и завершения учебных занятий по темам, относящимся к представленному практическому заданию. Выполнение практических заданий – важный этап в профессиональной подготовке слушателей, так как способствует повышению качества усвоения программного материала, углубленному пониманию наиболее сложных вопросов курса, овладению методами решения задач по программированию, а также совершенствованию профессиональных навыков и умений.

8. Оценка качества освоения дисциплины

Итоговая аттестация по дисциплине проводится в форме зачета.

В качестве оценочных средств используются:

- задания для самостоятельной работы (размещены в СДО Moodle в материалах лекционных и практических занятий),
- итоговое испытание (практическое задание-кейс).

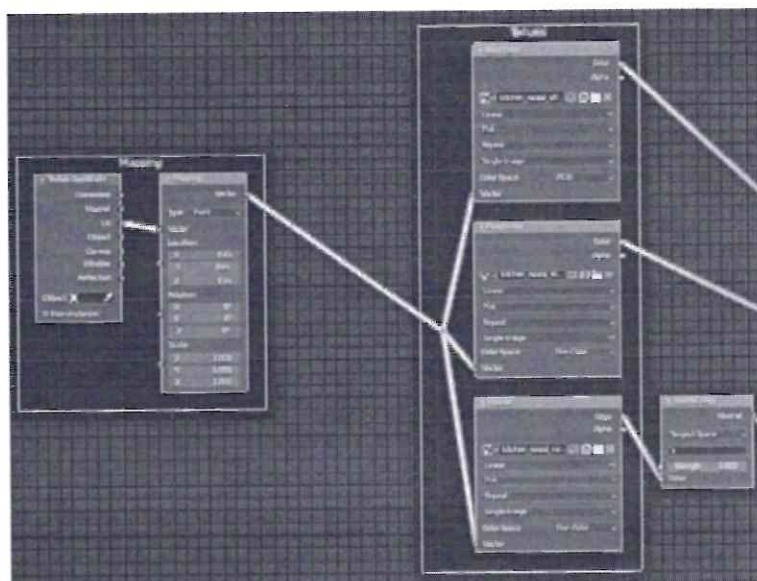
Задания для самостоятельной работы

Задание 1: PBR-материалы

- Посетите <https://polyhaven.com/textures> или <https://ambientcg.com/list?type=Material>
- Скачайте и распакуйте архив любого материала
- Подключите аддон **Node Wrangler**
- На объекте в редакторе шейдеров — выделите нод стандартного материала **Principled BSDF**
- Нажмите **Shift+Ctrl+T** для автоматического подключения карт, выберите нужные и подтвердите



Карты сами подцепятся как надо



Иногда не все карты подключаются

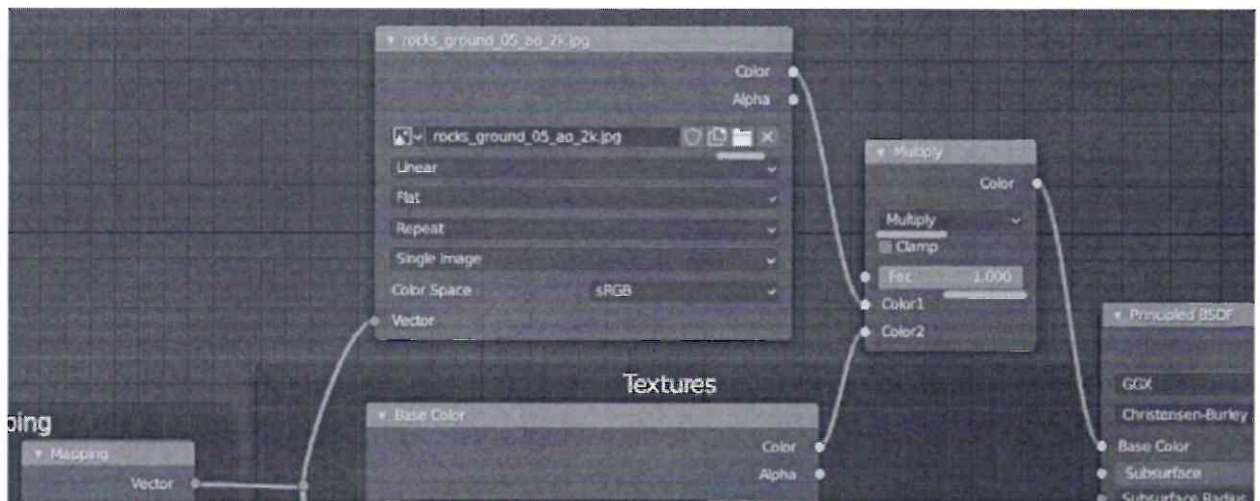
- Добавьте ноду для подключения файла
- Откройте карту с **AO (Ambient Occlusion)**

Add (Shift A) ▶ Texture ▶ Image Texture

- Добавьте ноду для смешивания

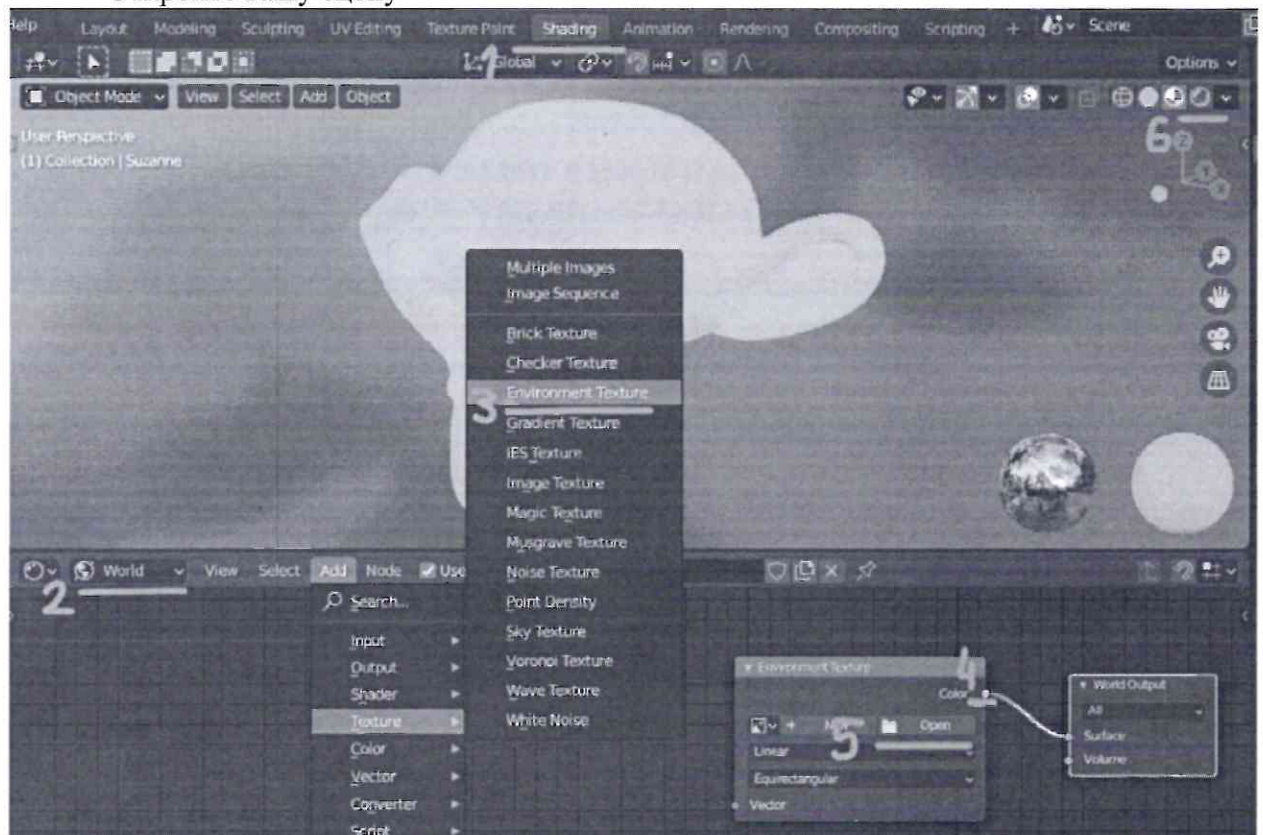
Add (Shift A) ▶ Color ▶ MixRGB

- Выберите параметр **Multiply**
- Выставьте фактор на 1.0
- Слейте карту **AO** с **Base Color** с помощью нода смешивания



Задание 2: HDR-карта окружения

- Посетите <https://polyhaven.com/hdris>
- Скачайте любую **HDR**-карту окружения
- Откройте вашу сцену



1. Перейдите на вкладку **Shading**
2. В редакторе шейдеров выберите **World**
3. Добавьте ноду **Environment Texture**
4. Соедините в соответствии со схемой
5. Откройте скачанную карту
6. Настройте вьюпорт на вывод рендера

Дополнительно

- Убедитесь что аддон **Node Wrangler** активен
- Нажмите **Ctrl+T** выбрав карту окружения
- Настройте карту окружения нодой **Mapping**

ФОС Итоговой аттестации по дисциплине

Кейс 1

- Используя примитивы соберите простую сцену, например снеговика и ёлку.
- Добавьте в сцену 3d-моделью с использованием полигонального моделирования, например дерево с ветвями.
- Вычислите и сохраните рендер.
- Приложите рендер и blend-файл проекта.

ПРОВЕРЯЕМЫЕ ЗНАНИЯ:

1. Объекты-примитивы (кубы, цилиндры, конусы, сферы, торы)
2. Полигональное моделирование
3. Трансформации
4. Освещение
5. Материалы
6. Модификаторы
7. Настройка камеры
8. Рендеринг

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ «АНИМАЦИЯ ПЕРСОНАЖЕЙ»

1. Цель дисциплины: знакомство с анимацией, способами и возможностями ее создания и настройки; формирование необходимых умений и навыков для профессиональной деятельности в области анимации персонажей.

2. Планируемые результаты обучения, соотнесенные с формируемыми компетенциями:

Компетенция	Планируемые результаты обучения
ПК-5: Использует 3д моделирование	Знать: основы полигонального моделирования, процесс создания развертки и текстуринга моделей. Уметь: работать с программой 3D моделирования Blender; создавать элементы графического пользовательского интерфейса; создавать 3D-модели, анимации и сцены для приложений и игр, выполнять рендеринг модели или сцены. Владеть навыками: использования инструментов среды моделирования Blender, использования модификаторов, материалов и света.

3. Трудоемкость дисциплины

Общая трудоемкость дисциплины составляет 36 ч. По дисциплине предусмотрены лекции в объеме 14 часов, практические занятия – 18 часов, СРС – 4 ч. и ИКСР слушателя – 2 ч на слушателя. Форма итогового контроля: зачет.

4. Содержание дисциплины

№, Наименование темы	Содержание лекций (количество часов)	Наименование лабораторных работ (количество часов)	Наименование практических занятий или семинаров (количество часов)	Виды СРС (количество часов)
1	2	3	4	5
Скелетная анимация	6	0	6	0
Лицевая анимация	4	0	6	2
Камера, свет, рендеринг	4	0	6	2
Всего 36	14	0	18	4

5. Учебно-методическое обеспечение СРС

Согласно действующему Учебному плану на самостоятельную работу слушателя по дисциплине «Анимация персонажей» отводится 4 часа. СРС по данной дисциплине предполагает подготовку к текущему и итоговому контролю, выполнение практических заданий.

Самостоятельная работа предусматривает изучение отдельных разделов и тем учебной дисциплины, изучение программного обеспечения, подготовку к зачету, выполнение заданий.

Текущая и опережающая СРС:

- работа с лекционным материалом, поиск и анализ литературы и электронных источников информации по отдельным темам;
- подготовка к практическим занятиям, выполнение домашних заданий;
- изучение тем, вынесенных на самостоятельную проработку;
- выполнение практических заданий;
- подготовка к итоговой аттестации.

К каждой теме приведены контрольные задания. Для самостоятельной подготовки предложен теоретический материал, непосредственно связанный с выполняемыми практическими работами, а также список рекомендуемой литературы и источников в сети интернет.

Форма контроля текущей и опережающей СРС – устный опрос, проверка выполнения практических заданий, задание-кейс.

6. Организационно-педагогические условия реализации дисциплины:

а) Материально-технические условия

Режим реализации: очно-заочная, с применением дистанционных образовательных технологий и электронного обучения.

Минимальные требования к компьютеру:

- Intel Core i3 3240 (или аналог от AMD) и выше.
- Оперативная память DDR3 и выше объемом 8 Гб и более.
- Видеокарта с объемом памяти от 2 Гб и выше (для разработки приложений допускается использовать встроенную видеокарту).
- Наличие от 50Гб свободного места на SSD или HDD.

- Монитор (или экран ноутбука) с разрешением экрана 1440 x 900 точек и глубиной цвета 32 bit (рекомендуемое разрешение экрана 1920 x 1080).
- Акустическая система или наушники.
- Доступ в Интернет со скоростью от 10 Мбит/с и выше.

Информационные технологии:

- Мультимедиа-технологии.
- Интернет-технологии (наличие выхода в интернет с целью поиска современной практико-ориентированной и учебной литературы, а также учебных видеокурсов по дисциплине).
- Дистанционные технологии обучения на базе СДО.

Программное обеспечение:

- Операционная система Windows 10.
- Интернет-браузер Яндекс.
- Программа для 3D моделирования Blender.
- OBS Studio – бесплатная программа с открытым исходным кодом для записи видео и потокового вещания.
- Discord – кроссплатформенная проприетарная система мгновенного обмена сообщениями с поддержкой VoIP и видеоконференций.
- Система дистанционного обучения Moodle.

б) Учебно-методическое и информационное обеспечение

Основная литература:

1. Хэсс Ф. Практическое пособие Blender 3.0 для любителей и профессионалов. Моделинг, анимация, VFX, видеомонтаж. – М.: Солон-Пресс. 2022. – 300 с.
2. Создание персонажей в Blender и Unity. [Электронный ресурс] – URL: <https://habr.com/ru/post/422255/>

Дополнительная литература:

1. Слаква А. Инструменты моделирования в Blender. [Электронный ресурс] – URL: <https://disk.yandex.ru/d/p8Sm9Eods24y0Q>
2. Blender Secrets Vol. 1 + 2 + 3. [Электронный ресурс] – URL: <https://online-courses.club/blender-secrets-vol-1-2-3-ebook/>

Перечень ресурсов сети Интернет:

1. Создаем анимацию реконструкции объекта в Blender. [Электронный ресурс] – URL: <https://dtf.ru/gamedev/714977-sozdaem-animaciyu-rekonstrukcii-obekta-v-blender-e>
2. Анимация и риггинг. [Электронный ресурс] – URL: <https://blender3d.com.ua/category/animation/>
3. Blender 3.1 Reference Manual. [Электронный ресурс] – URL: <https://docs.blender.org/manual/en/latest/index.html>
4. Официальный сайт Blender. – URL: <https://www.blender.org/>
5. Бесплатные текстуры Poly Haven. [Электронный ресурс] – URL: <https://polyhaven.com/textures>
6. Бесплатные текстуры ambientCG. [Электронный ресурс] – URL: <https://ambientcg.com/list>

7. Методические указания для обучающихся

К каждой теме дисциплины приведены упражнения для закрепления материала. Для самостоятельной подготовки предложен теоретический материал, непосредственно связанный с выполняемыми работами, а также список рекомендуемой литературы и источников в сети интернет.

Передача необходимых теоретических знаний и формирование основных представлений по каждой теме дисциплины происходит с использованием лекционных занятий. Лекционный материал закрепляется в ходе практических занятий. Изучение дисциплины сопровождается самостоятельной работой слушателей в форме изучения пройденного лекционного материала, подготовки к практическим занятиям и выполнения индивидуальных практических заданий в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков обучающихся.

Если самостоятельно не удастся разобраться в материале, необходимо сформулировать вопрос и задать преподавателю на занятии.

Необходимо уделить внимание следующим понятиям: создание частиц, исправление меша, скелетная анимация, риггинг.

При реализации различных видов учебной работы также используются:

- Электронные лекции в СДО Moodle.
- Электронные образовательные ресурсы (как предлагаемые интернет, так и авторские).

Практические задания являются одним из видов самостоятельной работы слушателей. Они выполняются после прослушивания лекций и завершения учебных занятий по темам, относящимся к представленному практическому заданию. Выполнение практических заданий – важный этап в профессиональной подготовке слушателей, так как способствует повышению качества усвоения программного материала, углубленному пониманию наиболее сложных вопросов курса, овладению методами решения задач по программированию, а также совершенствованию профессиональных навыков и умений.

9. Оценка качества освоения дисциплины

Итоговая аттестация по дисциплине проводится в форме зачета.

В качестве оценочных средств используются:

- задания для самостоятельной работы (размещены в СДО Moodle в материалах лекционных и практических занятий),
- итоговое испытание (практическое задание-кейс).

Задания для самостоятельной работы

Задание 1: Скелетная анимация

- Добавьте персонажу костную систему с помощью сервиса mixamo.
- Разработайте пример скелетной анимации, пусть персонаж меняет положение частей тела в пространстве, садится или прыгает, например.

Задание 2: Лицевая анимация

- С помощью ключей формы создайте персонажу измененное выражение лица, помимо базового.
- Разработайте пример лицевой анимации, пусть персонаж меняет выражение лица от базового к измененному, например.

Задание 3: Простая анимация и анимация параметров

- Разработайте пример анимации, когда объект меняет положение, вращение, масштаб, например движение камеры, источников света, любых объектов в кадре.
- Разработайте пример анимации параметров, например у камеры меняется фокусное расстояние или у источника света меняется цвет.

Кейс 2

Нужны примеры выполнения разных видов анимации, желательно всё в 1 ролике в формате **mp4**:

- Скелетная анимация – персонаж меняет положение частей тела в пространстве, прыгает или садится, например.
- Лицевая анимация – у персонажа меняется выражение лица.
- Простая анимация: положение, вращение, масштаб – движение объектов в кадре, анимация камеры.
- Анимация параметров – например изменение цветов источников света, фокусного расстояния камеры.

Если не достаточно места для вашей анимации, то можно разместить ссылкой в облаке, но в таком случае просим не удалять файлы из облака несколько месяцев.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ «РАЗРАБОТКА 3D ИГР И ПРИЛОЖЕНИЙ НА UNITY»

1. Цель дисциплины: знакомство с основами разработки 3D игр и приложений с использованием игрового движка Unity на языке C#, формирование необходимых умений и навыков для выполнения профессиональных задач.

2. Планируемые результаты обучения, соотнесенные с формируемыми компетенциями:

Компетенция	Планируемые результаты обучения
ПК-1: Применяет стандарты и методики при оформлении программного кода	<p>Знать: внутреннюю организацию программного кода, правила и рекомендации оформления программного кода, стандарты и соглашения о написании программного кода на C#.</p> <p>Уметь: использовать стандарты и соглашения при написании программного кода, оптимально и эффективно организовывать программный код.</p> <p>Владеть навыками: навыками оформления программного кода в соответствии с принятыми правилами и стандартами.</p>
ПК-3: Применяет языки программирования	<p>Знать: синтаксис языка программирования C#, особенности программирования на этом языке, пакеты программ в области программирования (Microsoft Visual Studio Community, Unity); общие принципы и навыки практического применения объектно-ориентированного программирования на C#.</p> <p>Уметь: проектировать и разрабатывать 3D приложения и игры для ПК, проводить их отладку и тестирование; разрабатывать интерфейс пользователя для игр и приложений; проводить рефакторинг и оптимизацию программного кода; использовать возможности имеющейся технической и/или программной архитектуры.</p> <p>Владеть навыками: методами использования в профессиональной деятельности языка программирования C#; инструментами игрового движка Unity; современными методами практического программирования; умениями и навыками использования библиотек объектов (классов) игрового движка Unity для решения практических задач.</p>

ПК-4: Применяет интегрированные среды разработки (IDE)	<p>Знать: основы языка программирования C# и основы работы с IDE (Microsoft Visual Studio).</p> <p>Уметь: использовать одну или несколько современных IDE для написания программного кода.</p> <p>Владеть навыками: использования IDE для разработки приложений виртуальной реальности в соответствии с поставленной задачей.</p>
--	--

3. Трудоемкость дисциплины

Общая трудоемкость дисциплины составляет 72 ч. По дисциплине предусмотрены лекции в объеме 18 часов, практические занятия – 48 часов, СРС – 6 ч. и ИКСР слушателя – 2 ч на слушателя. Форма итогового контроля: зачет.

4. Содержание дисциплины

№, Наименование темы	Содержание лекций (количество часов)	Наименование лабораторных работ (количество часов)	Наименование практических занятий или семинаров (количество часов)	Виды СРС (количество часов)
1	2	3	4	5
Игровой движок Unity. Возможности на сегодняшний день	2	0	0	0
Архитектура проекта. Структура проекта в Unity. Работа со сценой	2	0	0	0
Процесс разработки игрового проекта в среде Unity	8	0	36	4
Создание элементарного интерфейса с использованием Canvas	6	0	12	2
Всего 72	18	0	48	6

5. Учебно-методическое обеспечение СРС

Согласно действующему Учебному плану на самостоятельную работу слушателя по дисциплине «Разработка 3D игр и приложений на Unity» отводится 6 часов. СРС по данной дисциплине предполагает подготовку к текущему и итоговому контролю, выполнение задания-кейса (проектная работа).

Самостоятельная работа предусматривает изучение отдельных разделов и тем учебной дисциплины, изучение возможностей программного обеспечения, подготовку к зачету, выполнение дополнительных заданий, полученных от преподавателя на занятиях.

Текущая и опережающая СРС:

- работа с лекционным материалом, поиск и анализ литературы и электронных источников информации по отдельным темам;
- подготовка к практическим занятиям, выполнение домашних заданий;
- изучение тем, вынесенных на самостоятельную проработку;
- выполнение проектного задания (кейс);

- подготовка к итоговой аттестации.

Для самостоятельной подготовки предложен теоретический материал, непосредственно связанный с выполняемыми практическими работами, а также список рекомендуемой литературы и источников в сети интернет.

Форма контроля текущей и опережающей СРС – устный опрос, тестирование, проверка домашнего задания, проверка выполнения проектного задания.

6. Организационно-педагогические условия реализации дисциплины:

а) Материально-технические условия

Режим реализации: очно-заочная, с применением дистанционных образовательных технологий и электронного обучения.

Минимальные требования к компьютеру:

- Intel Core i3 3240 (или аналог от AMD) и выше.
- Оперативная память DDR3 и выше объемом 8 Гб и более.
- Видеокарта с объемом памяти от 2 Гб и выше (для разработки приложений допускается использовать встроенную видеокарту).
- Наличие от 50Гб свободного места на SSD или HDD.
- Монитор (или экран ноутбука) с разрешением экрана 1440 x 900 точек и глубиной цвета 32 bit (рекомендуемое разрешение экрана 1920 x 1080).
- Акустическая система или наушники.
- Доступ в Интернет со скоростью от 10 Мбит/с и выше.

Информационные технологии:

- Мультимедиа-технологии.
- Интернет-технологии (наличие выхода в интернет с целью поиска современной практико-ориентированной и учебной литературы, а также учебных видеокурсов по дисциплине).
- Технологии компьютерного тестирования.
- Дистанционные технологии обучения на базе СДО.

Программное обеспечение:

- Операционная система Windows 10.
- Интернет-браузер Яндекс;
- Интегрированная среда разработки (IDE) Microsoft Visual Studio Community 2019 (2022).
- Игровой движок Unity 2020.2.7.
- OBS Studio – бесплатная программа с открытым исходным кодом для записи видео и потокового вещания.
- Discord – кроссплатформенная проприетарная система мгновенного обмена сообщениями с поддержкой VoIP и видеоконференций.
- Система дистанционного обучения Moodle.

б) Учебно-методическое и информационное обеспечение

Основная литература:

1. Торн А. Искусство создания сценариев в Unity: руководство / А. Торн; перевод с английского Р. Н. Рагимова. – Москва: ДМК Пресс, 2016. – 360 с. – Текст: электронный // Электронно-библиотечная система «Лань». – URL: <https://e.lanbook.com/book/82812>
2. Курбанисмаилов З.М. Современные подходы в программировании при создании интерактивной анимации на C# и Unity: учебно-методическое пособие / З.М. Курбанисмаилов. – Москва: РТУ МИРЭА, 2021. – 142 с. – Текст: электронный // Лань: электронно-библиотечная система. – URL: <https://e.lanbook.com/book/176569>

3. 2D игра на Unity. Подробное руководство. [Электронный ресурс]. – URL: <http://websketches.ru/blog/2d-igra-na-unity-podrobnoye-rukovodstvo-p1>

Дополнительная литература:

1. Дикинсон К. Оптимизация игр в Unity 5 / К. Дикинсон. – Москва: ДМК Пресс, 2017. – 306 с. – Текст: электронный // Электронно-библиотечная система «Лань». – URL: <https://e.lanbook.com/book/90109>
2. Кенни Л. Шейдеры и эффекты в Unity. Книга рецептов / Л. Кенни; под редакцией В.В. Симонова; перевод с английского Е.А. Шапочкин. – Москва: ДМК Пресс, 2014. – 274 с. – Текст: электронный // Электронно-библиотечная система «Лань». – URL: <https://e.lanbook.com/book/58687>

Перечень ресурсов сети Интернет:

1. Интерактивный учебник по C#. [Электронный ресурс]. – URL: <https://docs.microsoft.com/ru-ru/dotnet/csharp/programming-guide/>
2. Справочник по C#. [Электронный ресурс]. – URL: <https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/>
3. Unity C# уроки (краткий вводный курс). [Электронный ресурс]. – URL: <https://itproger.com/course/unity-csharp>
4. Сочетания клавиш Unity. [Электронный ресурс]. – URL: <https://www.sites.google.com/site/rusewyl/home/socetania-klavis-unity>
5. Сцепление с дорогой (wheel collider unity3d). [Электронный ресурс]. – URL: <https://docs.unity3d.com/ru/2018.4/Manual/class-WheelCollider.html>
6. Шпаргалка по Unity и краткий справочник. [Электронный ресурс]. – URL: <https://stdpub.com/unity3d/shpargalka-po-unity-i-kratkij-spravochnik>
7. Работа с Корутинами в Unity. [Электронный ресурс]. – URL: <https://habr.com/ru/post/216185/>
8. Введение в Unity2D. [Электронный ресурс]. – URL: <https://jwinters.ru/unity3d/introduction-to-unity2d/>
9. Процедурное создание уровней в Unity. Часть 2. Создание окружения из текстового файла. [Электронный ресурс]. – URL: <https://studhelper.blogspot.com/2019/12/procLevel-2.html>

7. Методические указания для обучающихся

К каждой теме дисциплины приведены упражнения для закрепления материала. Для самостоятельной подготовки предложен теоретический материал, непосредственно связанный с выполняемыми работами, а также список рекомендуемой литературы и источников в сети интернет.

Передача необходимых теоретических знаний и формирование основных представлений по каждой теме дисциплины происходит с использованием лекционных занятий. Лекционный материал закрепляется в ходе практических занятий. Изучение дисциплины сопровождается самостоятельной работой слушателей в форме изучения пройденного лекционного материала, подготовки к практическим занятиям и выполнения индивидуальных заданий-кейсов в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков обучающихся.

Если самостоятельно не удастся разобраться в материале, необходимо сформулировать вопрос и задать преподавателю на занятии.

Необходимо уделить внимание следующим понятиям: коллайдер, коллизия, триггер, слой, тэг, скрипт, шейдер, событие, спрайт, ассет, префаб, корутина.

При реализации различных видов учебной работы также используются:

– Электронные лекции в СДО Moodle.

– Электронные образовательные ресурсы (как предлагаемые интернет, так и авторские).

Проектная работа является одним из видов самостоятельной работы слушателей. Она выполняется после прослушивания лекций и завершения учебных занятий по всем темам, запланированным к изучению в рамках дисциплины. Выполнение проектной работы – важный этап в профессиональной подготовке слушателей, так как способствует повышению качества усвоения программного материала, углубленному пониманию наиболее сложных вопросов курса, овладению методами решения задач по программированию, а также совершенствованию профессиональных навыков и умений.

В ходе выполнения проектной работы по созданию 3D игры для ПК под управлением ОС Windows, слушатели закрепляют пройденный материал, учатся самостоятельно писать скрипты, создавать необходимые объекты из префабов, управлять физикой объектов, создавать интерфейс пользователя. По завершению работы над проектом выполняется тестирование приложения. По результатам тестирования принимается решение о доработке приложения, или его сдаче преподавателю.

8. Оценка качества освоения дисциплины

Итоговая аттестация по дисциплине проводится в форме зачета.

В качестве оценочных средств используются:

- устный опрос,
- тестирование,
- практические задания (для самоконтроля),
- итоговое испытание (кейс, проектная работа).

Вопросы к устному опросу

1. Что такое Unity3D. Возможности на сегодняшний день. Интерфейс.
2. Рабочие окна. Настройка рабочего пространства.
3. Работа со сценой.
4. Игровые объекты. Компоненты.
5. Камера сцены. Режимы. Настройка.
6. Архитектура проекта. Структуру проекта в Unity.
7. Понятие ассета, импорт и экспорт Package.
8. Asset Store. Работа во вкладке Project.
9. Понятие префаба.
10. Организация объектов. Тэги. Слои.
11. Ассеты. Виды, особенности, специфика.
12. Запуск и отладка сцены. Настройки проекта.
13. Импорт графики из 3D-редакторов. Работа с материалами.
14. Шейдеры, используемые в Unity3D. Текстуры и специальные карты.
15. Физика в Unity.
16. Настройка тел и их коллайдеров. Гравитация.
17. Физические материалы. Работа с движущимися объектами.
18. Создание сценариев на языке программирования C#.
19. Программирование логики для игровых объектов.
20. Взаимодействие между игровыми объектами на сцене.
21. События игрового объекта.
22. Взаимодействие с мышью, клавиатурой, экраном смартфона.

Тестирование

1. Какой язык программирования используется в Unity для написания скриптов?

1. C++
2. C#
3. Object Pascal
4. Fortran

2. Укажите (впишите) соответствия между элементами, указанными ниже:

1. Вызывается, когда текущий collider/rigidbody начал соприкосновение с другим rigidbody/collider	
2. Вызывается, когда коллайдер входит в триггер	
3. Уничтожает указанный GameObject	

Варианты для соответствия: OnCollisionEnter, OnTriggerEnter, Destroy

3. Чтобы произошло событие OnTriggerEnter, должны быть выполнены условия:

1. Имеется объект с коллайдером и компонентом Rigidbody
2. Второй взаимодействующий объект должен иметь коллайдер
3. Один из взаимодействующих коллайдеров должен быть помечен как триггер
4. Ни на каком из взаимодействующих объектов не должно быть коллайдеров

4. Unity поддерживает работу с моделями в формате *.blend (файл Blender 3D)?

1. Да
2. Нет.

5. Укажите (впишите) соответствия между представленными ниже словами и их описаниями:

1. Программный код, который предназначен для управления игровыми объектами и создания логики игры	
2. Выполняется один раз при запуске сцены, при каждом включении скрипта, при создании объекта с этим скриптом	
3. Выводит сообщение в консоль и записывает сообщение в отдельном лог-файле	

Варианты для соответствия: Скрипт (script), Метод Start(), Команда Debug.Log()

6. Когда выполняется метод Update() ?

1. Каждый раз при выходе из игры
2. Каждый раз при запуске игры
3. Каждый кадр
4. Один раз при запуске сцены

7. Что такое float?

1. Базовый числовой тип переменной с плавающей запятой, содержит число с десятичной дробной частью
2. Базовый числовой тип переменной, содержит целое число
3. Специальный тип объекта, контейнер для компонентов Unity
4. Специальный тип объекта, содержит координаты, поворот и масштаб игровых объектов Unity

8. Что такое Transform?

1. Определяет Mass (массу), Velocity (скорость), Drag (сопротивление) объектов в сцене. У каждого GameObject'а есть Transform.
2. Определяет Position (положение), Rotation (вращение), и Scale (масштаб) каждого объекта в сцене. У каждого GameObject'а есть Transform.
3. Определяет Color (цвет), Texture (текстуру), Shader (шейдер) объектов в сцене. Не у всех GameObject'ов есть Transform.
4. Определяет Color (цвет), Mode (режим), Intensity (интенсивность) освещения объектов в сцене. У каждого GameObject'а есть Transform.

9. Что такое Material?

1. Material это элемент Unity, определяет как расположен GameObject. Совокупность параметров координат, поворота, масштаба.
2. Material это элемент Unity, определяет как GameObject взаимодействует с физикой. Совокупность параметров массы, скорости.
3. Material это элемент Unity, определяет как отображается поверхность. Совокупность параметров, текстур и Shader'a.
4. Material это элемент Unity, определяет как работает GameObject. Совокупность компонентов.

10. Что такое Rigidbody?

1. Компонент Rigidbody определяет Color (цвет), Texture (текстуру), Shader (шейдер) объектов в сцене.
2. Компонент Rigidbody определяет Mass (массу), Velocity (скорость), Drag (сопротивление) и физическое взаимодействие между объектами.
3. Компонент Rigidbody определяет Color (цвет), Mode (режим), Intensity (интенсивность) освещения объектов в сцене.
4. Компонент Rigidbody определяет Position (положение), Rotation (вращение), и Scale (масштаб) каждого объекта в сцене.

11. Какое свойство отвечает за то, чтобы Коллайдер использовался для запуска событий, и игнорировался физическим движком?

1. Is Trigger
2. Direction
3. Step Offset
4. Friction

12. Установите соответствие (впишите) между компонентами и их назначением:

1. Определение физических границ столкновения для объектов	
2. Дает объекту физические свойства в реальном времени, такие как вес и гравитация	
3. Заставляет объект вести себя как источник света с различными эффектами	
4. Отвечает за рендеринг и отображение объектов	

Варианты для соответствия: Коллайдер, Rigidbody, Light, Renderer

13. Верно ли, что для того, чтобы создать экземпляр GameObject, нужно вызвать метод Instantiate() ?

1. Да
2. Нет.

14. Эта функция вызывается, когда этот collider/rigidbody начал соприкосновение с другим rigidbody/collider.

1. OnCollisionEnter
2. Start
3. Awake
4. OnStartCollision

15. Верно ли, что GameObjects может быть чем угодно, от модели игрока до графического интерфейса на экране, от кнопок и врагов до невидимых «менеджеров», таких как источники звука?

1. Да
2. Нет.

Практические задания (для самоконтроля)

1. Создать уровень, по которому можно будет свободно перемещаться, а также брать и при желании уничтожать объекты. Создать соответствующее по стилю уровня окружение.
2. Создать модульную автоматизированную дверь, в самом общем ее смысле, состоящую из нескольких мешей, выступающих в качестве дверной рамы и двери. Дверь реагирует на приближение игрока. Реализовать также возможность ввода кода для двери, по которому она будет открываться. Создать соответствующее по стилю уровня окружение.
3. Создать модель вращающегося огненного кубика, который начинает вращаться при приближении к нему игрока. Кубик после касания игроком переходит в пользование игрока, движется вместе с ним на некотором расстоянии, и может уничтожать предметы, оказывающиеся на пути игрока (например, подлетает к ним и уничтожает при касании). Создать соответствующее по стилю уровня окружение.

ФОС Итоговой аттестации по дисциплине

Кейс 3.

1. Реализовать модель квартала города, по дорогам которого перемещаются агенты (могут быть боты, автомобили, и т.д., на выбор). На перекрестках улиц работают видеочамеры, которые передают информацию на основной экран игрока. Игрок просто должен наблюдать за происходящим на улицах и иметь возможность отключать или включать соответствующие видео с камер.
2. Создать игру от первого лица, которая длится бесконечно. Для этого нужно выполнить реализацию следующих элементов (реализация – аналогично примеру с автомобилем в городе): бесконечно перемещать игрока вперед; генерировать препятствия, которые игрок должен избегать; рандомизировать препятствия для создания вариаций; создавать кнопку перезапуска, которая отображается, когда игрок сталкивается с препятствием; вести подсчет набранных игроком очков.
3. Имеется простая локация, содержащая объекты (в простом случае – в форме параллелепипедов), бота и игрока (в простом случае – обычные капсулы, различающиеся цветом). Бот характеризуется параметрами: скорость движения, радиус просмотра. Реализовать поведение бота, когда он может убежать от игрока и прятаться за препятствиями. Один из вариантов реализации такого поведения следующий. Бот на основе Raycast проверяет, есть ли между ним и игроком препятствие – если его нет, то в радиусе своей видимости он выбирает позицию, из которой Raycast на игрока будет давать препятствие. После этого бот начинает

перемещаться в эту позицию, параллельно проверяя положение игрока и при необходимости корректируя позицию. Перемещение игрока зависит от того, как реализована игра – в 3D или VR.

4. Сформировать небольшую локацию с соответствующим окружением. Локация должна содержать игрока и объект, позволяющий производить ботов (точка спавна). Точка спавна ботов представляет собой особое двухэтажное (или выше) здание, в нижней части которого будут появляться боты. Появление нового бота должно сопровождаться соответствующим эффектом (например, дым, огненный шар, и т.д.). После появления бот бежит до определенной точки на локации и исчезает (уничтожается). Перемещение бота должно использовать анимацию, можно использовать стандартные модели из сервиса Mixamo.
5. Написать прототип шутера от первого лица, в котором игроку требуется пройти миссию, выполнив некоторое задание: захватить точку (точки), уничтожить всех противников, найти артефакт (артефакты), собрать определенное количество предметов. Реализовать в игре:
 - Соответствующее окружение (строения, деревья и кустарники, трава, другие объекты для создания определенной атмосферы). *Для создания окружения можно использовать готовые ассеты, не содержащие программного кода (т.е. использовать в готовом виде можно только модели).*
 - Использование оружия игроком.
 - Взаимодействие с ботами, у вас в игре могут быть боты ближнего и дальнего боя (последние могут использовать определенный вид оружия).
 - Простую систему частиц для создания определенных эффектов (огня, дыма, использования магии и прочее).
 - Звуки выстрелов, ходьбы, и другие по необходимости.
 - Вывод статистики (количество набранных игроком очков, здоровье игрока, количество убитых противников, и т.д., в зависимости от геймплея).

В качестве расширения возможностей игры можно предложить:

- Реализовать систему прокачки игрока (например, увеличение брони, здоровья, атаки, подбор боеприпасов, и пр.).
- Ввести в игру летающих ботов. В качестве алгоритма управления такими ботами можно использовать пример моего проекта с полетом пчел по различным задаваемым траекториям.
- Реализовать БПЛА, которые будут следовать за игроком, и защищать его от атак ботов. Характеристики БПЛА, условия их появления у игрока, определить самостоятельно.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ «ПРОЕКТИРОВАНИЕ И АРХИТЕКТУРА ПРИЛОЖЕНИЙ VR»

1. Цель дисциплины: знакомство с основами разработки игр и приложений виртуальной реальности на основе SteamVR Plugin с использованием игрового движка Unity и языка программирования C#, формирование необходимых умений и навыков для выполнения профессиональных задач.

2. Планируемые результаты обучения, соотнесенные с формируемыми компетенциями:

Компетенция	Планируемые результаты обучения
ПК-1: Применяет стандарты и методики при оформлении программного кода	<p>Знать: внутреннюю организацию программного кода, правила и рекомендации оформления программного кода, стандарты и соглашения о написании программного кода на C#.</p> <p>Уметь: использовать стандарты и соглашения при написании программного кода, оптимально и эффективно организовывать программный код.</p> <p>Владеть: навыками оформления программного кода в соответствии с принятыми правилами и стандартами.</p>
ПК-2: Оценивает возможности применения виртуальной и дополненной реальностей	<p>Знать: историю появления и эволюцию технологии виртуальной реальности (VR), области применения технологии VR, программные и технические средства, использующиеся при работе с виртуальной реальностью.</p> <p>Уметь: использовать программные и технические средства для работы с виртуальной реальностью; разрабатывать приложения и игры, использующие виртуальную реальность.</p> <p>Владеть: представлениями о технологиях виртуальной и дополненной реальности, инструментами и SDK для разработки приложений виртуальной реальности.</p>
ПК-3: Применяет языки программирования	<p>Знать: синтаксис языка программирования C#, особенности программирования на этом языке, пакеты программ в области программирования (Microsoft Visual Studio Community, Unity); общие принципы и навыки практического применения объектно-ориентированного программирования на C#.</p> <p>Уметь: проектировать и разрабатывать 3D приложения и игры для ПК; разрабатывать игры и приложения виртуальной реальности на основе SteamVR Plugin с использованием игрового движка Unity и языка программирования C#, проводить их отладку и тестирование; разрабатывать интерфейс пользователя для игр и приложений; проводить рефакторинг и оптимизацию программного кода; использовать возможности имеющейся технической и/или программной архитектуры.</p> <p>Владеть: методами использования в профессиональной деятельности языка программирования C#; инструментами игрового движка Unity; современными методами практического программирования; умениями и навыками использования библиотек объектов (классов) игрового движка Unity для решения практических задач.</p>
ПК-4: Применяет интегрированные среды разработки (IDE)	<p>Знать: основы языка программирования C# и основы работы с IDE (Microsoft Visual Studio).</p> <p>Уметь: использовать одну или несколько современных IDE для написания программного кода.</p> <p>Владеть: навыками использования IDE для разработки приложений дополненной реальности в соответствии с поставленной задачей.</p>

3. Трудоемкость дисциплины

Общая трудоемкость дисциплины составляет 72 ч. По дисциплине предусмотрены лекции в объеме 18 часов, практические занятия – 48 часов, СРС – 6 ч. и ИКСР слушателя – 2 ч на слушателя. Форма итогового контроля: экзамен.

4. Содержание дисциплины

№, Наименование темы	Содержание лекций (количество часов)	Наименование лабораторных работ (количество часов)	Наименование практических занятий или семинаров (количество часов)	Виды СРС (количество часов)
1	2	3	4	5
Виртуальная реальность (VR). Оборудование для работы VR	2	0	0	0
Способы работы с VR	2	0	0	0
Создание проекта VR на Unity. Взаимодействие с объектами при работе с VR	8	0	36	4
Создание интерфейсов пользователя при работе с VR	6	0	12	2
Всего 72	18	0	48	6

5. Учебно-методическое обеспечение СРС

Согласно действующему Учебному плану на самостоятельную работу слушателя по дисциплине «Проектирование и архитектура приложений VR» отводится 6 часов. СРС по данной дисциплине предполагает подготовку к текущему и итоговому контролю, выполнение задания-кейса (проектная работа).

Самостоятельная работа предусматривает изучение отдельных разделов и тем учебной дисциплины, изучение возможностей программного обеспечения, подготовку к экзамену, выполнение дополнительных заданий, полученных от преподавателя на занятиях.

Текущая и опережающая СРС:

- работа с лекционным материалом, поиск и анализ литературы и электронных источников информации по отдельным темам;
- подготовка к практическим занятиям, выполнение домашних заданий;
- изучение тем, вынесенных на самостоятельную проработку;
- выполнение проектного задания (кейс);
- подготовка к итоговой аттестации.

Для самостоятельной подготовки предложен теоретический материал, непосредственно связанный с выполняемыми практическими работами, а также список рекомендуемой литературы и источников в сети интернет.

Форма контроля текущей и опережающей СРС – проверка домашнего задания, проверка выполнения проектного задания.

6. Организационно-педагогические условия реализации дисциплины:

а) Материально-технические условия

Режим реализации: очно-заочная, с применением дистанционных образовательных технологий и электронного обучения.

Минимальные требования к компьютеру:

- Intel Core i3 3240 (или аналог от AMD) и выше.
- Оперативная память DDR3 и выше объемом 8 Гб и более.
- Видеокарта с объемом памяти от 2 Гб и выше (для разработки приложений допускается использовать встроенную видеокарту).
- Наличие от 50Гб свободного места на SSD или HDD.
- Монитор (или экран ноутбука) с разрешением экрана 1440 x 900 точек и глубиной цвета 32 bit (рекомендуемое разрешение экрана 1920 x 1080).
- Акустическая система или наушники.
- Доступ в Интернет со скоростью от 10 Мбит/с и выше.

Информационные технологии:

- Мультимедиа-технологии.
- Интернет-технологии (наличие выхода в интернет с целью поиска современной практико-ориентированной и учебной литературы, а также учебных видеокурсов по дисциплине).
- Дистанционные технологии обучения на базе СДО.

Программное обеспечение:

- Операционная система Windows 10.
- Интернет-браузер Яндекс.
- Интегрированная среда разработки (IDE) Microsoft Visual Studio Community 2019 (2022).
- Игровой движок Unity 2020.2.7.
- OBS Studio – бесплатная программа с открытым исходным кодом для записи видео и потокового вещания.
- Discord – кроссплатформенная проприетарная система мгновенного обмена сообщениями с поддержкой VoIP и видеоконференций.
- Система дистанционного обучения Moodle.

б) Учебно-методическое и информационное обеспечение

Основная литература:

1. Джонатан Л. Виртуальная реальность в Unity / Л. Джонатан; перевод с английского Р.Н. Рагимов. – Москва: ДМК Пресс, 2016. – 316 с. – Текст: электронный // Лань: электронно-библиотечная система. – URL: <https://e.lanbook.com/book/93271>
2. Смолин А.А., Жданов Д.Д., Потемин И.С., Меженин А.В., Богатырев В.А. Системы виртуальной, дополненной и смешанной реальности Учебное пособие. – Санкт-Петербург: Университет ИТМО. 2018. – 59 с.

Дополнительная литература:

1. Сердюков Ю.М. Философия виртуальной реальности и искусственного интеллекта: учебное пособие / Ю.М. Сердюков; под редакцией Ю.М. Сердюкова. – Хабаровск: ДВГУПС, 2020. – 169 с. – Текст: электронный // Лань: электронно-библиотечная система. – URL: <https://e.lanbook.com/book/179385>

Перечень ресурсов сети Интернет:

1. Начните делать игры виртуальной реальности в Unity 5 бесплатно. [Электронный ресурс]. – URL: <http://helpexe.ru/igry/nachnite-delat-igry-virtualnoj-realnosti-v-unity-5>

2. Unity & Vive: Tutorial. [Электронный ресурс]. – URL: <https://habr.com/ru/post/426357/>
3. SteamVR Unity Plugin. [Электронный ресурс]. – URL: https://valvesoftware.github.io/steamvr_unity_plugin/?ref=https://githubhelp.com

7. Методические указания для обучающихся

К каждой теме дисциплины приведены упражнения для закрепления материала. Для самостоятельной подготовки предложен теоретический материал, непосредственно связанный с выполняемыми работами, а также список рекомендуемой литературы и источников в сети интернет.

Передача необходимых теоретических знаний и формирование основных представлений по каждой теме дисциплины происходит с использованием лекционных занятий. Лекционный материал закрепляется в ходе практических занятий. Изучение дисциплины сопровождается самостоятельной работой слушателей в форме изучения пройденного лекционного материала, подготовки к практическим занятиям и выполнения индивидуальных заданий-кейсов в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков обучающихся.

Если самостоятельно не удастся разобраться в материале, необходимо сформулировать вопрос и задать преподавателю на занятии.

Необходимо уделить внимание следующим понятиям: контроллеры, SteamVR Plugin, область телепортации, телепорт, взаимодействие с предметами в виртуальной реальности, алгоритм создания приложения VR.

При реализации различных видов учебной работы также используются:

- Электронные лекции в СДО Moodle.
- Электронные образовательные ресурсы (как предлагаемые интернет, так и авторские).

Проектная работа является одним из видов самостоятельной работы слушателей. Она выполняется после прослушивания лекций и завершения учебных занятий по всем темам, запланированным к изучению в рамках дисциплины. Выполнение проектной работы – важный этап в профессиональной подготовке слушателей, так как способствует повышению качества усвоения программного материала, углубленному пониманию наиболее сложных вопросов курса, овладению методами решения задач по программированию, а также совершенствованию профессиональных навыков и умений.

В ходе выполнения проектной работы должно быть создано приложение виртуальной реальности с использованием SteamVR Plugin. По результатам тестирования готового приложения принимается решение о его доработке или сдаче преподавателю.

Проектная работа, реализуемая в рамках данной дисциплины, дорабатывается и может быть представлена обучающимся к защите на итоговой аттестации.

8. Оценка качества освоения дисциплины

Итоговая аттестация по дисциплине проводится в форме зачета.

В качестве оценочных средств используются:

- практические задания (для самоконтроля),
- итоговое испытание (кейс, проектная работа).

Практические задания (для самоконтроля)

1. Создать уровень, по которому можно будет свободно перемещаться, а также брать и при желании уничтожать объекты. Создать соответствующее по стилю уровня окружение.

2. Имеется простая локация, содержащая объекты (в простом случае – в форме параллелепипедов), бота и игрока (в простом случае – обычные капсулы, различающиеся цветом). Бот характеризуется параметрами: скорость движения, радиус просмотра. Реализовать поведение бота, когда он может убежать от игрока и прятаться за препятствиями. Один из вариантов реализации такого поведения следующий. Бот на основе Raycast проверяет, есть ли между ним и игроком препятствие – если его нет, то в радиусе своей видимости он выбирает позицию, из которой Raycast на игрока будет давать препятствие. После этого бот начинает перемещаться в эту позицию, параллельно проверяя положение игрока и при необходимости корректируя позицию. Приложение должно быть реализовано в VR.
3. Создать клавиатуру для использования в VR-приложениях для ввода имени игрока. После ввода имени клавиатура должна скрываться. Такая клавиатура может быть использована в случае наступления определенных событий, например, при:
 - Первом входе игрока в игру.
 - Наборе игроком определенного количества очков и попадании в таблицу рекордов.
 - Организации диалога с виртуальным персонажем для ввода возможных ответов, или вопросов.

Также клавиатура (или ее отдельные функциональные элементы) может быть использована при разработке обучающего приложения для системы образования.

ФОС Итоговой аттестации по дисциплине

Кейс 5

1. Сформировать локацию для игры, в которой требуется бросать дротики в воздушные шарики, расположенные на некотором удалении от игрока в ячейках ящика. Реализовать систему обновления шариков и генерации дротиков. Приложение должно подсчитывать очки при попадании игроком по шарик (шарик при попадании должен лопнуть). Создать соответствующее окружение для локации. Игра должна содержать точку выхода.
2. Написать приложение для игры в боулинг. Сформировать соответствующую локацию, содержащую от одной до 3 дорожек. Мячи боулинга появляются автоматически на устройстве подачи мячей около начала дорожки. Кегли выставляются автоматически после удара мяча по ним. В процессе игры необходимо вести подсчет очков, набранных игроком. Игра должна содержать точку выхода.
3. Написать приложение для бросания баскетбольного мяча в корзину. Приложение должно быть реализовано в виде баскетбольной площадки. Игрок может бросать мяч с любой точки на площадке, в зависимости от дальности броска должны подсчитываться очки при попадании мяча в корзину. Попаданием считать только движение мяча в корзину сверху вниз, если мяч брошен снизу под корзиной, то такой бросок не засчитывается. Обеспечить отскок мяча несколько раз от поверхности после падения на пол площадки, после чего он должен появиться вблизи игрока. Игра должна содержать точку выхода.
4. Сформировать небольшую локацию с соответствующим окружением. Локация должна содержать игрока и расположенные на некотором расстоянии от него мишени. Примерный вид мишеней до попадания и после попадания пуль показан на рисунке справа. Игрок стреляет по мишени из пистолета, количество выстрелов не ограничено. После поражения всех мишеней они автоматически возвращаются в первоначальное состояние (первая



мишень на рисунке). Должна вестись статистика, включающая в себя: количество попаданий по мишеням, общее количество выстрелов, результативность (отношение количества попаданий к общему количеству выстрелов, отображать 3 знака после запятой). Статистика отображается на каком-либо соответствующим образом оформленном элементе, и должна быть хорошо видна игроку. Игра должна содержать точку выхода.

5. Написать приложение, локация которого представляет собой загородный дом с участком. План дома и участка разработать самостоятельно. Реализовать возможность перемещения игрока по участку, вход в дом и перемещение по комнатам. Двери комнат должны открываться автоматически при приближении игрока (расстояние реагирования двери на игрока подобрать самостоятельно). Обеспечить взаимодействие с игроком некоторых предметов в доме (предмет можно взять и переместить). Игра должна содержать точку выхода.

4. ОЦЕНКА КАЧЕСТВА ОСВОЕНИЯ ПРОГРАММЫ

Оценка качества освоения программы включает текущую, промежуточную и итоговую аттестацию обучающихся.

ИТОГОВАЯ АТТЕСТАЦИЯ СЛУШАТЕЛЕЙ

Итоговая аттестация слушателей проводится в форме защиты проектной работы. Разработка проектов обучающимися начинается на дисциплине «Проектирование и архитектура приложений VR» с выбора темы проектной работы и дальнейшей реализации проектной работы.

Проектная работа представляет собой программный продукт (приложение) для ПК, работающего под управлением ОС Windows 10 и написанный на языке программирования C# с использованием SteamVR Plugin.

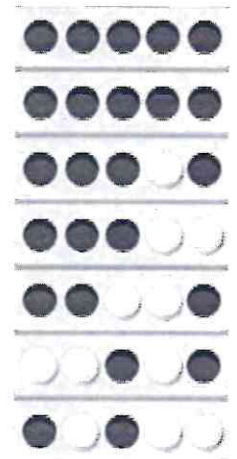
Итоговая аттестация является обязательной и осуществляется после освоения образовательной программы в полном объеме.

Итоговые аттестационные испытания предназначены для определения профессиональных компетенций слушателя, определяющих его подготовленность к решению профессиональных задач, установленных федеральным государственным образовательным стандартом и профессиональным стандартом, способствующих его устойчивости на рынке труда и продолжению образования на следующих ступенях.

Фонд оценочных средств для итоговой аттестации

1. Сформировать локацию для игры, в которой требуется бросать дротики в воздушные шарики, расположенные на некотором удалении от игрока в ячейках ящика. Реализовать систему обновления шариков и генерации дротиков. Приложение должно подсчитывать очки при попадании игроком по шарик (шарик при попадании должен лопнуть). Создать соответствующее окружение для локации. Игра должна содержать точку выхода.

2. Написать приложение для игры в боулинг. Сформировать соответствующую локацию, содержащую от одной до 3 дорожек. Мячи боулинга появляются автоматически на устройстве подачи мячей около начала дорожки. Кегли выставляются автоматически после удара мяча по ним. В процессе игры необходимо вести подсчет очков, набранных игроком. Игра должна содержать точку выхода.
3. Написать приложение для бросания баскетбольного мяча в корзину. Приложение должно быть реализовано в виде баскетбольной площадки. Игрок может бросать мяч с любой точки на площадке, в зависимости от дальности броска должны подсчитываться очки при попадании мяча в корзину. Попаданием считать только движение мяча в корзину сверху вниз, если мяч брошен снизу под корзиной, то такой бросок не засчитывается. Обеспечить отскок мяча несколько раз от поверхности после падения на пол площадки, после чего он должен появиться вблизи игрока. Игра должна содержать точку выхода.
4. Сформировать небольшую локацию с соответствующим окружением. Локация должна содержать игрока и расположенные на некотором расстоянии от него мишени. Примерный вид мишеней до попадания и после попадания пуль показан на рисунке справа. Игрок стреляет по мишени из пистолета, количество выстрелов не ограничено. После поражения всех мишеней они автоматически возвращаются в первоначальное состояние (первая мишень на рисунке). Должна вестись статистика, включающая в себя: количество попаданий по мишеням, общее количество выстрелов, результативность (отношение количества попаданий к общему количеству выстрелов, отображать 3 знака после запятой). Статистика отображается на каком-либо соответственным образом оформленном элементе, и должна быть хорошо видна игроку. Игра должна содержать точку выхода.



или включать соответствующие видео с камер. Приложение может быть выполнено как в 3D, так и в VR. Приложение должно содержать точку выхода.

Критерии оценивания проектной работы

- 36-40 баллов (оценка «отлично») выставляется слушателю, если он правильно выполнил задание, представленный проект не содержит ошибок, при проектировании использовались общепринятые нормы и принципы.
- 31-35 баллов (оценка «хорошо») выставляется слушателю, если он правильно выполнил задание, представленный проект не содержит ошибок, при проектировании использовались общепринятые нормы и принципы, но в проекте присутствуют некоторые недоработки (в целом не влияющие на основной функционал приложения, и позволяющие использовать приложение по его прямому назначению).
- 24-30 баллов (оценка «удовлетворительно») выставляется слушателю, если он представил в целом правильно выполненное задание, которое, однако, содержало отдельные ошибки; также при проектировании могли не использоваться общепринятые нормы и принципы; тем не менее, при этом приложение вполне работоспособно и в нем можно использовать хотя бы часть функционала.
- менее 24 баллов (оценка «не удовлетворительно») выставляется слушателю, если он представил частично выполненное задание (или он не приступал к выполнению задания), при его выполнении не использовались общепринятые нормы и принципы, приложение не работоспособно; представленный проект не соответствует требованию задачи.

**5. КАЛЕНДАРНЫЙ УЧЕБНЫЙ ПЛАН-ГРАФИК
ОСВОЕНИЯ ПРОГРАММЫ**

Наименование дисциплины	Объем нагрузки в час.	Учебные недели														
		1 неделя	2 неделя	3 неделя	4 неделя	5 неделя	6 неделя	7 неделя	8 неделя	9 неделя	10 неделя	11 неделя	12 неделя			
3D моделирование	72	24	24	20	4 ЗЧ											
Анимация персонажей	36				20	16 ЗЧ										
Разработка 3D игр и приложений на Unity	72					12		32	28 ЗЧ							
Проектирование и архитектура приложений VR	72									24	20	20	8 ЭКЗ			
Итоговая аттестация	4															4
	256	24	24	20	24	28	32	28	24	20	20	8				4

6. СОСТАВИТЕЛИ ПРОГРАММЫ

Дмитриев Владислав Леонидович, кандидат физико-математических наук, доцент, доцент кафедры прикладной информатики и программирования Стерлитамакского филиала Башкирского государственного университета.