

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГАОУ ВО «СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»



УТВЕРЖДАЮ:

Директор НОЦ «Институт
непрерывного образования

Е.В. Мошкина

« 19 » декабря 2022 г.

ДОПОЛНИТЕЛЬНАЯ ПРОФЕССИОНАЛЬНАЯ ПРОГРАММА
ПРОФЕССИОНАЛЬНОЙ ПЕРЕПОДГОТОВКИ

«Разработка мобильных игр на Unity»

Красноярск 2022

I. ОБЩАЯ ХАРАКТЕРИСТИКА ПРОГРАММЫ

1.1. Аннотация программы

Дополнительная профессиональная программа (программа профессиональной переподготовки) ИТ-профиля «Разработка мобильных игр на Unity» (далее — Программа) разработана в соответствии с нормами Федерального закона РФ от 29 декабря 2012 года № 273-ФЗ «Об образовании в Российской Федерации»; с учетом требований приказа Минобрнауки России от 1 июля 2013 г. № 499 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным профессиональным программам», с изменениями, внесенными приказом Минобрнауки России от 15 ноября 2013 г. № 1244 «О внесении изменений в Порядок организации и осуществления образовательной деятельности по дополнительным профессиональным программам, утвержденный приказом Министерства образования и науки Российской Федерации от 1 июля 2013 г. № 499»; приказа Министерства образования и науки РФ от 23 августа 2017 г. № 816 «Об утверждении Порядка применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ»; паспорта федерального проекта «Развитие кадрового потенциала ИТ-отрасли» национальной программы «Цифровая экономика Российской Федерации»; постановления Правительства Российской Федерации от 13 мая 2021 г. № 729 «О мерах по реализации программы стратегического лидерства «Приоритет-2030» (в редакции постановления Правительства Российской Федерации от 14 марта 2022 г. № 357 «О внесении изменений в постановление Правительства Российской Федерации от 13 мая 2021 г. № 729»); приказа Министерства цифрового развития, связи и массовых коммуникаций Российской Федерации от 28 февраля 2022 г. № 143 «Об утверждении методик расчета показателей федеральных проектов национальной программы «Цифровая экономика Российской Федерации» и признании утратившими силу некоторых приказов Министерства цифрового развития, связи и массовых коммуникаций Российской Федерации об утверждении методик расчета показателей федеральных проектов национальной программы «Цифровая экономика Российской Федерации»; федерального государственного образовательного стандарта высшего образования по направлению подготовки 09.03.02 Информационные системы и технологии (уровень бакалавриата), утвержденного приказом Минобрнауки России от 19 сентября 2017 г. № 926, (далее — ФГОС ВО), а также профессионального стандарта 06.001 «Программист», утвержденного приказом Министерства труда и социальной защиты РФ от 18 ноября 2013 г. № 679н.

Профессиональная переподготовка заинтересованных лиц (далее — Слушатели), осуществляемая в соответствии с Программой, имеющей отраслевую направленность «Информационно-коммуникационные технологии», проводится в ФГАОУ ВО «Сибирский федеральный университет» (далее — Университет) в соответствии с учебным планом в очно-заочной форме обучения.

Разделы, включенные в учебный план Программы, используются для последующей разработки календарного учебного графика, учебно-тематического плана, рабочих программ модулей (дисциплин), оценочных и методических материалов. Перечисленные документы разрабатываются Университетом самостоятельно, с учетом актуальных положений законодательства об образовании, законодательства в области информационных технологий и смежных областей знаний ФГОС ВО и профессионального стандарта 06.001 «Программист».

За последние годы смартфоны прочно вошли в жизнь большинства людей. Индустрия мобильных приложений претерпевает бурное развитие и все больше предпринимателей нуждаются в разработчиках. Крупным корпорациям мобильные приложения нужны с целью автоматизации бизнес-процессов, в том числе, в части взаимодействия персонала и клиентов, получению статистической отчетности. Мобильные приложения используются также в сфере электронной коммерции. С их помощью можно моментально обрабатывать запросы клиента, рассказывать об акциях, развивать систему лояльности и постоянно напоминать о себе с помощью пуш-уведомлений. После прохождения обучения слушатель сможет самостоятельно разрабатывать мобильные приложения, используя популярный движок Unity.

1.2. Цель программы

Цель программы ДПП — формирование у слушателей, обучающихся по специальностям и направлениям подготовки, не отнесенным к ИТ-сфере, согласно приложению к Методике расчета показателя «Количество принятых на обучение по программам высшего образования в сфере информационных технологий за счет бюджетных ассигнований федерального бюджета (нарастающим итогом, начиная с 2021 года)», утвержденной приказом Минцифры России от 28 февраля 2022 г. № 143, цифровых компетенций в области информационных технологий, а именно разработка программного обеспечения, а также приобретение по итогам прохождения программы ДПП новой квалификации «Программист».

Целевая группа: слушатели, относящиеся к категории обучающихся по специальностям и направлениям подготовки, не отнесенным к ИТ-сфере.

1.3. Характеристика новой квалификации и связанных с ней видов профессиональной деятельности, трудовых функций и(или) уровней квалификации

1.3.1. Область профессиональной деятельности слушателя, прошедшего обучение по программе профессиональной переподготовки, в которой может осуществлять профессиональную деятельность: разработка программного обеспечения.

Выпускники могут осуществлять профессиональную деятельность в других областях и(или) сферах профессиональной деятельности при условии

соответствия уровня их образования и полученных компетенций требованиям к квалификации работника.

1.3.2. Объекты профессиональной деятельности: программное обеспечение, мобильные приложения.

Виды профессиональной деятельности: Разработка программного обеспечения.

1.3.3. Уровень квалификации. В соответствии с приказом Министерства труда и социальной защиты Российской Федерации от 12 декабря 2016 г. № 30635 «Об утверждении Профессионального стандарта «Программист», дополнительная профессиональная программа профессиональной переподготовки «Разработка мобильных игр на Unity» обеспечивает достижение шестого уровня квалификации.

1.3.4. Компетенции (трудовые функции) в соответствии с профессиональным стандартом (формирование новых или совершенствование имеющихся)

Программа разработана в соответствии с актуальными квалификационными требованиями, профессиональными стандартами специалистов. Виды профессиональной деятельности, трудовые функции, указанные в профессиональном стандарте 06.001 «Программист», представлены в таблицах 1–2.

Характеристика новой квалификации, связанной с видом профессиональной деятельности и трудовыми функциями в соответствии с профессиональным стандартом 06.001 «Программист»

Трудовые действия	Трудовая функция	Обобщенная трудовая функция	Вид профессиональной деятельности
<p>Составление формализованных описаний решений поставленных задач в соответствии с требованиями технического задания или других принятых в организации нормативных документов.</p> <p>Разработка алгоритмов решения поставленных задач в соответствии с требованиями технического задания или других принятых в организации нормативных документов</p>	<p>A/01.3 Формализация и алгоритмизация поставленных задач</p>	<p>A Разработка и отладка программного кода</p>	<p>Разработка программного обеспечения</p>
<p>Создание программного кода в соответствии с техническим заданием (готовыми спецификациями)</p>	<p>A/02.3 Написание программного кода с использованием языков программирования, определения и манипулирования данными</p>		
<p>Проверка работоспособности программного обеспечения на основе разработанных тестовых наборов данных</p>	<p>B/03.4 Проверка работоспособности программного обеспечения</p>	<p>B Проверка работоспособности и рефакторинг кода программного обеспечения</p>	
<p>Разработка, изменение и согласование архитектуры программного обеспечения с системным аналитиком и архитектором программного обеспечения</p>	<p>D/03.6 Проектирование программного обеспечения</p>	<p>D Разработка требований и проектирование программного обеспечения</p>	

Характеристика новой и развиваемой цифровой компетенции в ИТ-сфере, связанной с уровнем формирования и развития в результате освоения программы «Разработка мобильных игр на Unity»

Наименование сферы	Наименование профессиональной компетенции	Пример инструментов	0 – способность не проявляется/ проявляется в степени, недостаточной для отнесения к 1 уровню сформированности компетенции	1 – способность проявляется под внешним контролем / при внешней постановке задачи/ обучающийся пользуется готовыми, рекомендованными продуктами	2 – способность проявляется, но обучающийся эпизодически прибегает к экспертной консультации/ самостоятельно подбирает и пользуется готовыми продуктами	3 – способность проявляется системно / обучающийся модифицирует способность под определенные задачи / создает новый продукт, обучает других
Средства программной разработки	Применяет языки программирования для решения профессиональных задач	C#		+	–	–

1.4. Планируемые результаты обучения

Слушатели в результате освоения программы профессиональной переподготовки «Разработка мобильных игр на Unity» смогут:

РО1. Применять принципы и основы алгоритмизации.

РО2. Применять выбранные языки программирования для написания программного кода.

РО3. Разрабатывать процедуры проверки работоспособности программного обеспечения.

РО4. Использовать существующие типовые решения и шаблоны проектирования программного обеспечения.

РО5. Выполнять работы по созданию мобильных приложений с использованием игрового движка Unity.

РО6. Составлять ценностное предложение, анализировать рынок мобильных предложений и клиентский сегмент, выявлять конкурентные преимущества проекта.

1.5. Категория слушателей

Лица, получающие высшее образование по очной (очно-заочной) форме, лица, освоившие основную профессиональную образовательную программу (далее — ОПОП ВО) бакалавриата, в объеме не менее первого курса (бакалавры 2-го курса), ОПОП ВО специалитета — не менее первого и второго курсов (специалисты 3-го курса), обучающиеся по ОПОП ВО, НЕ отнесенным к ИТ-сфере.

1.6. Требования к уровню подготовки поступающего на обучение

В соответствии с требованиями к образованию и обучению, предъявляемыми к 6 уровню квалификации профессионального стандарта 06.001 «Программист», необходимо иметь высшее образование или осваивать его в момент обучения на данной программе.

1.7. Продолжительность обучения

256 часов, из них 128 контактных, в т.ч. 16 часов стажировка.

1.8. Форма обучения

Очно-заочная (обучение по программе реализовано в формате смешанного обучения, с применением электронного обучения и дистанционных образовательных технологий).

1.9. Требования к материально-техническому обеспечению, необходимому для реализации дополнительной профессиональной программы профессиональной переподготовки (требования к аудитории, компьютерному классу, программному обеспечению)

Обучение производится на платформе электронного обучения СФУ «е-Курсы» (<https://e.sfu-kras.ru/>). Используются сервисы вебинаров и видеоконференций.

При проведении лекций, практических занятий, самостоятельной работы слушателей и стажировки используется следующее оборудование: компьютер с наушниками или аудиоколонками, микрофоном и веб-камерой, высокоскоростное подключение к Интернет (не менее 5 Мбит/с).

Программное обеспечение (обновленное до последней версии): браузер Google Chrome, Microsoft Visual Studio, GNU GPL v.2. <https://git-scm.com/about/free-and-open-source>, Blender (GNU General Public License), Unity Education Grant.

1.10. Особенности (принципы) построения дополнительной профессиональной программы профессиональной переподготовки

Особенности построения программы переподготовки «Разработка мобильных игр на Unity»:

- в основу проектирования программы положен компетентностный подход;
- выполнение учебных заданий, требующих практического применения знаний и умений, полученных в ходе изучения логически связанных дисциплин;
- выполнение итоговых аттестационных работ по реальному заданию;
- использование информационных и коммуникационных технологий, в том числе современных систем технологической поддержки процесса обучения, обеспечивающих комфортные условия для обучающихся, преподавателей;
- применение электронных образовательных ресурсов (дистанционное, электронное, комбинированное обучение и пр.).

В поддержку дополнительной профессиональной программы профессиональной переподготовки разработан электронный курс: <http://e.sfu-kras.ru/course/view.php?id=29988>.

1.11. Особенности организации стажировки

Стажировка слушателей дополнительной профессиональной программы переподготовки «Разработка мобильных игр на Unity» является обязательной составной частью образовательной программы и представляет собой вид учебной деятельности, непосредственно ориентированный на профессионально-практическую подготовку слушателей. Стажировка осуществляется в целях формирования и закрепления профессиональных умений и навыков, полученных в результате теоретической подготовки.

Сроки проведения стажировки устанавливаются графиком учебного процесса в объеме 16 часов в конце процесса обучения в соответствии с утвержденным в установленном порядке учебно-тематическим планом.

В рамках очно-заочной формы обучения на основе дистанционных технологий стажировка осуществляется в форме online стажировки (в формате разработки игрового мобильного приложения на языке #).

1.12. Документ об образовании: диплом о переподготовке установленного образца.

УЧЕБНЫЙ ПЛАН
дополнительной профессиональной программы профессиональной переподготовки
«Разработка мобильных игр на Unity»

Форма обучения – очно-заочная.

Срок обучения – 256 часов.

№ п/п	Наименование дисциплин	Общая трудоемкость, ч	Всего контактн., ч	Контактные часы			СРС, ч	Формы контроля
				Лекции	Лабораторные работы	Практические и семинарские занятия		
1.	Введение в разработку игр	84	42	14		28	42	Зачет
2.	Программирование на языке с#	72	36	12		24	36	Зачет
3.	Разработка мобильного приложения на движке Unity	60	30	10		20	30	Зачет
4.	Стажировка	16	12			12	4	Зачет
5.	Итоговая аттестация	24	8			8	16	Защита итоговой аттестационной работы (проекта)
	Итого	256	128	36		92	128	

УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН
дополнительной профессиональной программы профессиональной переподготовки
«Разработка мобильных игр на Unity»

Категория слушателей: лица, имеющие или получающие высшее образование.

Срок обучения: 256 часов.

Форма обучения: очно-заочная.

Режим занятий: 6 часов в неделю.

№ п/п	Наименование дисциплин	Общая трудоемкость, ч	Всего контактн., ч	Контактные часы			СРС, ч	Результаты обучения
				Лекции	Лабораторные работы	Практ. и семинарские занятия		
1	Программирование на языке с#	84	42	14		28	42	PO1–PO4
1.1	Введение в язык с#. IDE, переменные и типы данных	12	6	2		4	6	PO1–PO4
1.2	Массивы и коллекции	12	6	2		4	6	PO1–PO4
1.3	Функциональное программирование	12	6	2		4	6	PO1–PO4
1.4	Объектно-ориентированное программирование	12	6	2		4	6	PO1–PO4
1.5	Наследование	12	6	2		4	6	PO1–PO4
1.6	Интерфейсы	12	6	2		4	6	PO1–PO4
1.7	Делегаты и события	12	6	2		4	6	PO1–PO4
2	Разработка мобильных игр на Unity	72	36	12		24	36	PO1–PO5
2.1	Введение в Unity	12	6	2		4	6	PO1–PO5
2.2	Написание программного кода в Unity	12	6	2		4	6	PO1–PO5
2.3	Физические взаимодействия	12	6	2		4	6	PO1–PO5
2.4	Визуальный компонент	12	6	2		4	6	PO1–PO5
2.5	Эстетика приложения	12	6	2		4	6	PO1–PO5
2.6	Публикация проекта	12	6	2		4	6	PO1–PO5
3.	Маркетинг и продвижение мобильного приложения	60	30	10		20	30	PO6
3.1	Виды технологического бизнеса и интернет-предпринимательство	12	6	2		4	6	PO6
3.2	Анализ рынка	12	6	2		4	6	PO6

№ п/п	Наименование дисциплин	Общая трудоем- кость, ч	Всего контактн., ч	Контактные часы			СРС, ч	Результаты обучения
				Лекции	Лабора- торные работы	Практ. и семинарские занятия		
3.3	Экономическая составляющая проекта	12	6	2		4	6	PO6
3.4	Продвижение продукта	12	6	2		4	6	PO6
3.5	Ресурсы и инвестиции	12	6	2		4	6	PO6
6	Стажировка	16	12			12	4	PO1–PO6
7	Итоговая аттестация	24	8	-		8	16	PO1–PO6
	Всего	256	128	32		96	128	

II. ОЦЕНКА КАЧЕСТВА ОСВОЕНИЯ ПРОГРАММЫ

2.1. Формы аттестации, оценочные материалы, методические материалы

Программа предусматривает проведение текущей и итоговой аттестации. Текущая аттестация слушателей проводится по дисциплинам на основе выполнения заданий в электронном обучающем курсе, а также с учетом результатов промежуточного ассесмента.

Методические материалы, необходимые для выполнения текущих заданий, представлены в соответствующих элементах электронного обучающего курса и включают описание задания, методические рекомендации по его выполнению, критерии оценивания.

2.2. Требования и содержание итоговой аттестации

К итоговой аттестации допускаются слушатели, выполнившие учебный план программы, самостоятельные задания в каждой дисциплине и успешно прошедшие процедуру итогового ассесмента. Итоговая аттестация по программе включает защиту итоговой аттестационной работы (ИАР) в форме проекта, которая проходит в синхронном формате.

Основная цель итоговой аттестационной работы — выполнить работу, демонстрирующую уровень подготовленности к самостоятельной профессиональной деятельности.

ИАР может выполняться как индивидуально, так и в группах 2-4 человека. Защита ИАР включает презентацию работы, вопросы по различным разделам программы. Защита ИАР дает возможность продемонстрировать уровень приобретенных слушателем профессиональных компетенций.

Примерный перечень тем работ:

1. Проектирование и разработка интерактивных учебных игр с применением среды Unity
2. Разработка игрового проекта на основе Unity
3. Разработка мобильной игры в жанре «Аркада» на платформе Unity.
4. Кооперативная игра в трехмерном лабиринте на Unity
5. Проектирование 2D-игр.
6. Программное игровое приложение с использованием мультиплатформенной инструментальной среды Unity.

Как правило, итоговая аттестационная работа состоит из 3–5 разделов (глав):

- обзор литературы по теме работы, в котором должны быть освещены различные точки зрения по затронутым в работе дискуссионным вопросам и обязательно сформулировано авторское отношение к ним;
- характеристика объекта и предмета разработки;
- характеристика методов выполнения работы;
- характеристика результатов выполнения работы и их интерпретация;

В заключении формулируются конкретные выводы по работе и предложения по их реализации.

Слушатель предоставляет результат выполненной работы в формате PDF, оформленной и отвечающей требованиям к содержанию итоговой аттестационной работы. Список использованных источников литературы приводится в конце ИАР. Документ прикрепляется в организационный электронный курс программы профессиональной переподготовки «Разработка мобильных игр на Unity». Объем презентации следует выбирать исходя из длительности выступления (обычно — не более 5–7 минут). В выступлении должны быть четко обозначены область и актуальность работы, постановка задачи, приведены результаты, полученные слушателем. Требования и содержание итоговой аттестации изложены в методических указаниях к выполнению ИАР и размещаются на платформе электронных курсов СФУ.

Критерии оценивания итоговой аттестационной работы

Оценка «отлично» ставится, если слушатель при выполнении ИАР применил язык программирования C# системно на экспертном уровне. Контролировал весь цикл программной разработки игрового мобильного проекта.

Оценка «хорошо» ставится, если слушатель при выполнении ИАР самостоятельно разработал отдельные модули.

Оценка «удовлетворительно» ставится, если слушатель при выполнении ИАР под контролем преподавателя разработал программное обеспечение.

Итоговая аттестационная работа защищается в синхронном формате перед аттестационной комиссией; работа представляется с помощью устного доклада и демонстрации презентации.

Защита итоговой аттестационной работы является обязательной.

Требования к устному докладу в режиме синхронной защиты

1. Приветствие, обращение к членам комиссии и представление.
2. Тема итоговой аттестационной работы.
3. Актуальность, цель и задачи работы.
4. Анализ результатов работы.
5. Заключение.

Продолжительность выступления — 7–8 минут.

По результатам защиты ИАР аттестационная комиссия принимает решение о присвоении слушателям по результатам освоения дополнительной профессиональной программы профессиональной переподготовки квалификации «Программист», о предоставлении права заниматься профессиональной деятельностью в сфере разработки программного обеспечения и выдаче диплома о профессиональной переподготовке.

III. ОСНОВНОЕ СОДЕРЖАНИЕ ПРОГРАММЫ

3.1. План учебной деятельности

Результаты обучения	Учебные действия/ формы текущего контроля	Используемые ресурсы/ инструменты/технологии
РО1. Применять принципы и основы алгоритмизации	Лекции. Выполнение задания, разработка алгоритмов решений в соответствии с требованиями технического задания. Тесты	Материалы электронного курса в системе электронного обучения СФУ «е-Курсы». Видеоконференции
РО2. Применять выбранные языки программирования для написания программного кода	Лекции. Выполнение задания, включающего этапы создания программного кода в соответствии с техническим заданием. Тесты	Материалы электронного курса в системе электронного обучения СФУ «е-Курсы». Видеоконференции
РО3. Разрабатывать процедуры проверки работоспособности программного обеспечения	Лекции. Выполнение задания, включающего этапы анализа и проверки исходного программного кода. Тесты	Материалы электронного курса в системе электронного обучения СФУ «е-Курсы». Видеоконференции
РО4. Использовать существующие типовые решения и шаблоны проектирования программного обеспечения	Лекции. Выполнение задания, включающего использование типовых решений и шаблонов проектирования программного обеспечения. Тесты	Материалы электронного курса в системе электронного обучения СФУ «е-Курсы». Видеоконференции
РО5. Выполнять работы по созданию мобильных приложений с использованием игрового движка Unity	Лекции. Выполнение задания, включающего использование типовых решений и шаблонов проектирования программного обеспечения. Тесты	Материалы электронного курса в системе электронного обучения СФУ «е-Курсы». Видеоконференции
РО6. Составлять ценностное предложение, анализировать рынок мобильных предложений и клиентский сегмент, выявлять конкурентные преимущества проекта.	Лекции. Выполнение задания, включающего использование типовых решений и шаблонов проектирования программного обеспечения. Тесты	Материалы электронного курса в системе электронного обучения СФУ «е-Курсы». Видеоконференции

3.2. Виды и содержание самостоятельной работы

Самостоятельная работа слушателя (СРС) предполагает углубление и закрепление теоретических знаний. СРС включает следующие виды самостоятельной деятельности: самостоятельное углубленное изучение вопросов программы, выполнение индивидуальных заданий, подготовка к тестированию и приобретение опыта работы в рамках электронного курса. Выполнение СРС предполагается в дистанционном режиме в рамках электронного курса.

РАБОЧАЯ ПРОГРАММА

дисциплины (модуля)

«Программирование на языке с#»

1. Аннотация

В рамках дисциплины «Программирование на языке с#» слушатели познакомятся с принципами решения задач в рамках методологии объектно-ориентированного программирования на языке с#, сформируют навыки решения прикладных задач, овладеют навыками использования инструментальных средств и различных технологий программирования, изучат методы проектирования, анализа и создания программных продуктов.

Цель дисциплины (результаты обучения)

По окончании обучения на данной дисциплине слушатели будут способны:

PO1. Применять принципы и основы алгоритмизации.

PO2. Применять выбранные языки программирования для написания программного кода.

PO3. Разрабатывать процедуры проверки работоспособности программного обеспечения.

PO4. Использовать существующие типовые решения и шаблоны проектирования программного обеспечения.

2. Содержание

№, наименование темы	Содержание лекций (кол-во часов)	Наименование практических (семинарских занятий) (кол-во часов)	Виды СРС (кол-во часов)
Модуль 1. Программирование на языке с# (72 часа)			
1.1. Введение в язык с#. IDE, переменные и типы данных (12 ч.)	Особенности программирования на языке с#. Среда разработки (2 ч.)	Интегрированная среда разработки (6 ч.). <i>Задание 1.</i> Переменные и типы данных. Создание первого приложения в Visual Studio (4 ч.)	Изучение IDE. Изучение теоретических материалов. Тестирование (6 ч.)
1.2. Массивы и коллекции (12 ч.)	Массивы и коллекции в с# (2 ч.)	Применение массивов и коллекций в с#. Управляющие конструкции в с# (4 ч.). <i>Задание 2.</i> Массивы и коллекции в с#	Операции в с#. Изучение теоретических материалов. Тестирование (6 ч.)

№, наименование темы	Содержание лекций (кол-во часов)	Наименование практических (семинарских занятий) (кол-во часов)	Виды СРС (кол-во часов)
1.3. Функциональное программирование (12 ч.)	Процедурное и функциональное программирование в с# (2 ч.)	Функциональное программирование (4 ч.). <i>Задание 3.</i> Использование функций в программе	Изучение встроенных библиотек в с#. Изучение теоретических материалов. Тестирование (6 ч.)
1.4. Объектно-ориентированное программирование (12 ч.)	Классы и объекты. Члены классов (2 ч.)	Объектно-ориентированное программирование в с# (4 ч.). <i>Задание 4.</i> Создание классов в с#	Инкапсуляция. Изучение теоретических материалов. Тестирование (6 ч.)
1.5. Наследование (12 ч.)	Наследование в с# (2 ч.)	Построение иерархии классов (4 ч.). <i>Задание 5.</i> Использование механизма наследования в программе	Полиморфизм. Раннее и позднее связывание. Изучение теоретических материалов. Тестирование (6 ч.)

3. Условия реализации программы дисциплины

Организационно-педагогические условия реализации программы

Обучение по программе реализовано в формате смешанного обучения, с применением активных технологий совместного обучения в электронной среде (синхронные и асинхронные занятия). Лекционный материал представляется в виде синхронных лекций, записей занятий, текстовых материалов, презентаций, размещаемых в электронном курсе. Данные материалы сопровождаются заданиями и дискуссиями в чатах дисциплин. Изучение теоретического материала (СРС) предполагается до и после синхронной части работы.

Материально-технические условия реализации программы

Синхронные занятия реализуются на базе инструментов видеоконференцсвязи и включают в себя лекционные и практические занятия. Для проведения синхронных занятий (вебинаров со спикерами) применяется программа видеоконференцсвязи. При проведении лекций, практических занятий, самостоятельной работы слушателей используется следующее оборудование: компьютер с наушниками или аудиоколонками, микрофоном и веб-камерой. Программное обеспечение (обновленное до последней версии): браузер Google Chrome, текстовый редактор.

Учебно-методическое и информационное обеспечение программы

Дисциплина может быть реализована как очно, так и заочно, в том числе, с применением дистанционных образовательных технологий. Она включает занятия лекционного типа, интерактивные формы обучения, практические занятия.

Содержание комплекта учебно-методических материалов

По данной дисциплине имеется электронный учебно-методический комплекс (УМК) в системе электронных курсов СФУ. УМК содержит: систему навигации по дисциплине (учебно-тематический план, интерактивный график работы по дисциплине, сведения о результатах обучения, чат для объявлений и вопросов преподавателю), текстовые материалы к лекциям, практические и тестовые задания, списки основной и дополнительной литературы. В электронном курсе реализована система обратной связи.

Литература

Основная литература

1. Семакин И.Г. и др. Программирование, численные методы и математическое моделирование. – М., 2021. – 288 с.
2. Иванова Г.С. Основы программирования: учеб. для вузов. – М., 2012.
3. Тузовский А. Объектно-ориентированное программирование: учеб. пособие для прикладного бакалавриата. – Litres, 2022.

Перечень ресурсов информационно-телекоммуникационной сети Интернет, необходимых для освоения дисциплины

1. Документация по с# [Электронный ресурс]. Режим доступа: <https://docs.microsoft.com/ru-ru/dotnet/csharp/>.

4. Оценка качества освоения программы дисциплины (формы аттестации, оценочные и методические материалы)

Форма аттестации по дисциплине — зачет.

Оценка результатов обучения осуществляется следующим образом. Максимально за курс можно набрать 100%, из них:

- тесты самоконтроля к лекциям 40 %;
- практические задания составляют 60 %.

Зачет получают слушатели, набравшие не менее 50 % из 100 от общего прогресса по курсу.

Примеры тестовых заданий для контроля знаний

Пример тестового задания по типу «Множественный выбор»

1. Какое ключевое слово необходимо применить в классе наследнике для переопределения виртуального метода базового класса?

- а) override;
- б) abstract;
- в) public;
- г) private.

Типовое практическое задание

Тема «Объектно-ориентированное программирование в C#»

Вам необходимо разработать виртуальный кошелек для студентов «Потратить стипендию разумно», который позволит контролировать траты. Информационная система «Потратить стипендию разумно» должна включать класс «Студент ИКИТа» StudentOfISIT. Необходимо хранить информацию о следующих характеристиках:

- ФИО студента (свойство string Name).
- Специальность обучения (свойство string Speciality).
- Размер ежемесячной стипендии (свойство int ScholarshipAmount).
- Виртуальный счет (свойство int Check).
- А также свойство, значение которого позволит отслеживать критически малый остаток на счету (меньше 100 р.) (свойство bool Warning). Данное свойство неразрывно связано с переменной Check. Значение данного свойства вычисляется в зависимости от остатка денег на счету, поэтому это свойство должно содержать только акцессор get.

Приход и расход средств осуществляется с использованием функций:

- void GetAScholarship() – функция, которая увеличивает размер счета на величину ScholarshipAmount, в случае, если текущая дата равна 20 числу. Проверку организовать с использованием DateTime.Now.Day.

- void SpendAScholarship(int money, string itemOfExpenditure) – функция, которая уменьшает размер счета на количество money, также необходимо указать статью расходов itemOfExpenditure. Например, SpendAScholarship(200, “поел в буфете”). После каждой транзакции пользователю на консоль должно выводиться сообщение о том сколько денег он потратил и на что. Также в функции SpendAScholarship необходимо предусмотреть проверку на условие: если переменная Warning равна true, то есть на счету больше 100 рублей, то студент может осуществлять траты, естественно в пределах этой суммы. Если же значение переменной равно false, то на экран должна выводиться фраза: «Денег на счету нет!».

Взаимодействие пользователя и информационной системы организуй через обращение к меню консоли (организуй через конструкцию switch/case). В меню должны быть пункты:

1. Задать имя студенту.

После того, как пользователь нажал «1» необходимо задать имя и специальность студенту. Спроси у пользователя его имя и специальность и проинициализируй соответствующие поля. В дальнейшем данная информация не подлежит изменению. То есть имя и специальность можно задать только 1 раз! Твое консольное приложение также должно позволять узнавать остаток на счете, но при этом размер ежемесячной стипендии задается только при инициализации объекта и меняться после уже не может Реализуй это через Init-only Properties and Object Initializers.

Примечание!

Реализовать одноразовое заполнение имени нужно на уровне самого свойства. Это значит, что вы можете в консоли дважды выбрать первую функцию и заполнить данные заново, но это не должно принести никакого эффекта.

Касательно `init`. Данный оператор доступен только в новой версии Visual Studio. Если вы работаете на более старой версии, вам может быть недоступна эта функциональность. В таком случае реализовать установление стипендии по аналогии с данными полного имени.

2. Ввести информацию о студенте

В этом пункте меню выведи на консоль информацию о имени студента, его специальности и размере ежемесячной стипендии.

3. Получить деньги

В этом пункте меню вызови функцию `GetAScholarship()`. Не забудь проверить, что сегодня действительно 20 число!

4. Потратить деньги

Спроси у пользователя сколько денег и на что он хочет их потратить и вызови функцию `SpendAScholarship`. Не забудь проверить, что у студента достаточное количество денег на счету!

5. Узнать, сколько денег на счету.

Здесь необходимо вывести на консоль значение свойства `Check`.

Критерии оценивания заданий

Баллы	1 балл	2 балла	3 балла
Критерий	Задание выполнено частично, требует серьезной доработки	Задание выполнено, но требует некоторой доработки	Задание выполнено полностью, не требует доработки

РАБОЧАЯ ПРОГРАММА
дисциплины (модуля)
«Разработка мобильного приложения на движке Unity»

1. Аннотация

Модуль «Разработка мобильного приложения на движке Unity» посвящен практическим аспектам разработки мобильных проектов с использованием движка Unity. Рассматриваются вопросы корректной установки необходимого ПО, навигации в редакторе, работа с объектами и компонентами, аспекты графики, скриптинг.

Цель дисциплины (результаты обучения)

По окончании обучения на данной дисциплине слушатели будут способны:

РО1. Применять принципы и основы алгоритмизации.

РО2. Применять выбранные языки программирования для написания программного кода.

РО3. Разрабатывать процедуры проверки работоспособности программного обеспечения.

РО4. Использовать существующие типовые решения и шаблоны проектирования программного обеспечения.

РО5. Выполнять работы по созданию мобильных приложений с использованием игрового движка Unity.

2. Содержание

№, наименование темы	Содержание лекций (кол-во часов)	Наименование практических (семинарских занятий) (кол-во часов)	Виды СРС (кол-во часов)
Модуль 2. Разработка мобильного приложения на движке Unity (72 часа)			
Тема 2.1. Введение в Unity (12 ч.)	Настройка рабочего места (Unity hub, версии Unity, установка, подключение VS). Интерфейс (основные окна) (2 ч.)	Подготовка рабочего места. Интерфейс (4 ч.). <i>Задание 1.</i> Основы работы в Unity – работа со сценой, компонент Transform	Asset store. Расширения редактора. Изучение теоретических материалов. Тестирование (6 ч.)
Тема 2.2. Написание программного кода в Unity (12 ч.)	Скриптинг. Использование классов Unity в коде. Работа со множеством сцен (2 ч.)	Создание сцены, инспектора. Создание скриптов в Unity (4 ч.). <i>Задание 2.</i> Создание скрипта для подгрузки сцены и перехода между сценами.	Ввод пользователя. События Unity. Работа с данными (PlayerPrefs + ScriptableObject). Изучение теоретических материалов. Тестирование (6 ч.)

№, наименование темы	Содержание лекций (кол-во часов)	Наименование практических (семинарских занятий) (кол-во часов)	Виды СРС (кол-во часов)
		Создание скрипта управления для персонажа игрока	
Тема 2.3. Физические взаимодействия (12 ч.)	Лучи. Физика (2 ч.)	Интерактивные сцены в юнити (4 ч.) <i>Задание 3.</i> Создание интерактивной сцены с использованием лучей	Физический движок Nvidia PhysX. Изучение теоретических материалов. Тестирование (6 ч.)
Тема 2.4. Визуальный компонент (12 ч.)	UI (компонент rectTransform компоновка) Графика (материалы и освещение) (2 ч.)	Интерфейс и освещение (4 ч.) <i>Задание 4.</i> Создание макета интерфейса. Настройка освещения на сцене	Старая система пользовательского интерфейса в Unity - GUI и её применение в Custom Inspector и Editor Window. Изучение теоретических материалов. Тестирование (6 ч.)
Тема 2.5. Эстетика приложения (12 ч.)	Анимации Работа со звуком (2 ч.)	Анимация и звук. (2 ч.). <i>Задание 5.</i> Создание макета интерфейса. Настройка освещения на сцене	Инверсная кинематика в анимации. Изучение теоретических материалов. Тестирование (6 ч.)
Тема 2.6 Публикация проекта (12 ч.)	Создание билда (2 ч.)	Создание билда (4 ч.) <i>Задание 6.</i> Создание билда тестового проекта и запуск на своей машине. Анализ проекта	Площадки для публикации проектов. Изучение теоретических материалов. Тестирование (6 ч.)

3. Условия реализации программы дисциплины

Организационно-педагогические условия реализации программы

Обучение по программе реализовано в формате смешанного обучения, с применением активных технологий совместного обучения в электронной среде (синхронные и асинхронные занятия). Лекционный материал представляется в виде синхронных лекций, записей занятий, текстовых материалов, презентаций, размещаемых в электронном курсе. Данные материалы сопровождаются заданиями и дискуссиями в чатах дисциплин. Изучение

теоретического материала (СРС) предполагается до и после синхронной части работы.

Материально-технические условия реализации программы

Синхронные занятия реализуются на базе инструментов видеоконференцсвязи и включают в себя лекционные и практические занятия. Для проведения синхронных занятий (вебинаров со спикерами) применяется программа видеоконференцсвязи. При проведении лекций, практических занятий, самостоятельной работы слушателей используется следующее оборудование: компьютер с наушниками или аудиокolonками, микрофоном и веб-камерой. Программное обеспечение (обновленное до последней версии): браузер Google Chrome, Microsoft Visual Studio, GNU GPL v.2. <https://git-scm.com/about/free-and-open-source>, Blender (GNU General Public License), Unity Education Grant.

Учебно-методическое и информационное обеспечение программы

Дисциплина может быть реализована как очно, так и заочно, в том числе, с применением дистанционных образовательных технологий. Она включает занятия лекционного типа, интерактивные формы обучения, практические занятия.

Содержание комплекта учебно-методических материалов

По данной дисциплине имеется электронный учебно-методический комплекс (УМК) в системе электронных курсов СФУ. УМК содержит: систему навигации по дисциплине (учебно-тематический план, интерактивный график работы по дисциплине, сведения о результатах обучения, чат для объявлений и вопросов преподавателю), текстовые материалы к лекциям, практические и тестовые задания, списки основной и дополнительной литературы. В электронном курсе реализована система обратной связи.

Литература

Основная литература

1. Торн А. Основы анимации в Unity. – М.: ДМК Пресс, 2016.
2. Торн А. Искусство создания сценариев в Unity. – М.: ДМК Пресс, 2016.
3. Суворов А.В., Медведков В.В., Саблина Г.В., Шахтштейндер В.Г. Программирование технологических контроллеров в среде Unity: учеб. пособие. – Новосибирск: НГТУ, 2016.

Перечень ресурсов информационно-телекоммуникационной сети Интернет, необходимых для освоения дисциплины

1. Unity User Manual 2020.3 (LTS) [Электронный ресурс]. – Режим доступа: <https://docs.unity3d.com/Manual/>.
2. Scripting API [Электронный ресурс]. – Режим доступа: <https://docs.unity3d.com/ScriptReference/>.

3. Welcome to Unity Learn [Электронный ресурс]. – Режим доступа: <https://learn.unity.com/>.

4. Оценка качества освоения программы дисциплины (формы аттестации, оценочные и методические материалы)

Форма аттестации по дисциплине — зачет.

Оценка результатов обучения осуществляется следующим образом. Максимально за курс можно набрать 100%, из них:

- тесты самоконтроля к лекциям 40 %;
- практические задания составляют 60 %.

Зачет получают слушатели, набравшие не менее 50 % из 100 от общего прогресса по курсу.

Примеры тестовых заданий для контроля знаний

Пример тестового задания по типу «Множественный выбор»

1. Для чего используется окно Project (окно Проекта)?
 - а) Позволяет настроить проект для компиляции на конкретную платформу.
 - б) Позволяет управлять всеми ассетами вашего проекта.
 - в) Позволяет посмотреть настройки проекта

Типовое практическое задание

Тема «Интерактивные сцены в Unity»

1. Создать скрипт **ObstacleItem**, в котором будет поле *currentValue*, отвечающее за «здоровье» этого препятствия (значение от 0 до 1) и поле *onDestroyObstacle* типа **UnityEvent**. Метод *GetDamage(float value)* будет отнимать от *currentValue* переданное значение. При этом, если *currentValue* = 1, цвет препятствия белый, при 0 — красный. Должен быть плавный переход цвета. Когда *currentValue* доходит до нуля, срабатывает *onDestroyObstacle* и сам объект **ObstacleItem** удаляется.

2. Создать скрипт **InteractiveBox**, который имеет метод *AddNext(InteractiveBox box)*, с помощью которого вы можете сохранять в поле *next* этого скрипта ссылку на другой **InteractiveBox**. У каждого в один момент времени может быть только один *next*, или не быть его вовсе. Если был добавлен *next*, вы должны испускать луч от текущего объекта до позиции объекта *next*. Отрисуйте этот луч с помощью **Debug** так, чтобы его было видно и в окне **Game** во время **playmode**. При этом, если луч попадает в объект с компонентом **ObstacleItem**, то для этого объекта будет вызываться метод *GetDamage(Time.DeltaTime)*. То есть, урон будет распределён по времени. (По факту, уничтожение произойдёт за 1 секунду, если у препятствия было полное «здоровье»).

3. Создать скрипт **InteractiveRaycast**, у которого будет поле *prefab* типа **GameObject**. В данном скрипте с помощью луча из камеры отслеживайте клики:

- Левый клик по любой поверхности объекта с тегом *InteractivePlane* создаёт в этой точке экземпляр объекта *prefab* (сам

префаб является кубиком со скриптом `InteractiveBox`). Учитывайте размер создаваемого объекта. Используйте `RaycastHit.normal`, чтобы избежать размещения объекта в текстурах.

- Левый клик по объекту с `InteractiveBox` запоминает этот компонент в локальной переменной скрипта. А левый клик по любому другому `InteractiveBox` добавляет уже сохранённому этот в качестве `next` с помощью метода `AddNext`.
- Правый клик по объекту с `InteractiveBox` удаляет его.

4. Создать сцену, где продемонстрировать работу скриптов. Создайте плоскость с тэгом `InteractivePlane`, разместите на ней несколько `ObstacleItem`, подключите к ним префабы из прошлой практической работы. Имеется ввиду добавление префабов со скриптами на сцену и подключение их метода `Use` как обработчиков в событиях `OnDestroyObstacle`. В итоге как только препятствие будет испускать событие, ваш префаб сработает, а само препятствие удалиться. При запуске вы сможете размещать на плоскости экземпляры `InteractiveBox` и связывать их, включая лучи. С помощью лучей вы сможете удалять `ObstacleItem`, которые будут запускать работу префабов в момент удаления.

Команда разработчиков должна сама распределять работу и решить, сколько должно быть веток и как не допустить конфликтов. Логика достаточно сильно связана. Каждый последующий скрипт требует наличия предыдущего. Вы можете создать программные интерфейсы, обеспечивающие связь. Также можно сначала создать “скелет” скриптов, и от него начать ветки с наполнением их логикой. Конфликты вероятны, но постарайтесь их избежать. Также воздержитесь от последовательного выполнения работы в одной ветке. Распараллелите её. Обращаем внимание, что есть много мелких моментов, которые могут привести к ошибкам. Уделите время тестированию. Для этого не бойтесь создавать логику для тестирования. Если вам нужно отследить работу луча, программно создайте сразу связанную пару. и т.д.

Критерии оценивания заданий

Баллы	1 балл	2 балла	3 балла
Критерий	Задание выполнено частично, требует серьезной доработки	Задание выполнено, но требует некоторой доработки	Задание выполнено полностью, не требует доработки

РАБОЧАЯ ПРОГРАММА
дисциплины (модуля)
«Маркетинг и продвижение мобильного приложения»

1. Аннотация

Целью настоящего модуля является формирование у студентов целостного представления о создании мобильного приложения в современных реалиях, получение теоретических знаний о принципах построения ИТ-проекта, основных его этапов развития от идеи до конечного продукта, разработки ценностного предложения, а также о создании презентации для демонстрации результатов работы над проектом.

Цель дисциплины (результаты обучения)

По окончании обучения на данной дисциплине слушатели будут способны:
 РОб. Составлять ценностное предложение, анализировать рынок мобильных предложений и клиентский сегмент, выявлять конкурентные преимущества проекта.

2. Содержание

№, наименование темы	Содержание лекций (кол-во часов)	Наименование практических (семинарских занятий) (кол-во часов)	Виды СРС (кол-во часов)
Модуль 3. Маркетинг и продвижение мобильного приложения (60 часов)			
3.1 Виды технологического бизнеса и интернет-предпринимательство (6 ч.)	Введение. Виды технологического бизнеса и интернет-предпринимательство. Идея: источник для создания проекта. Команда проекта (2 ч.)	Идея проекта. Команда и роли в команде (4 ч.). <i>Задание 1.</i> Формулирование и обоснование идеи проекта. Создание команды проекта и распределение ролей	Выделение клиентов и работа с ними. Создание ценностного предложения. Изучение теоретических материалов. Тестирование (6 ч.)
3.2 Анализ рынка (6 ч.)	Целевая аудитория. Анализ рынка. Оценка потенциала рынка. Ценностное предложение. Клиентское развитие, поиск и изучение клиентов. Анализ конкурентов (2 ч.)	Анализ конкурентов проекта, формулирование конкурентного преимущества. Расчет рынка (4 ч.). <i>Задание 2.</i> Маркетинговый анализ	Интервьюирование клиентов, анализ полученных результатов. Изучение теоретических материалов. Тестирование (6 ч.)
3.3 Экономическая составляющая проекта (6 ч.)	Модели монетизации. Метрики проекта и экономики продукта.	Выбор модели монетизации.	Ключевые метрики проекта.

№, наименование темы	Содержание лекций (кол-во часов)	Наименование практических (семинарских занятий) (кол-во часов)	Виды СРС (кол-во часов)
	Финансы проекта. Бизнес-модель (2 ч.)	Прогнозирование доходов проекта. <i>Задание 3.</i> Расчет ключевых метрик проекта	Изучение теоретических материалов. Тестирование (6 ч.)
3.4 Продвижение продукта (6 ч.)	Маркетинговые коммуникации. Постановка продаж. PR проекта (2 ч.)	Воронка продаж. <i>Задание 4.</i> Создание воронки продаж	Тестирование каналов и подготовка к масштабированию. Изучение теоретических материалов. Тестирование (6 ч.)
3.5 Ресурсы и инвестиции (6 ч.)	Инвестиции. Источники инвестиций. Подготовка презентации для инвестора (2 ч.)	Инвестиции проекта. Презентация для инвестора. <i>Задание 5.</i> Презентация проекта	Виды инвесторов. Изучение теоретических материалов. Тестирование (6 ч.)

3. Условия реализации программы дисциплины

Организационно-педагогические условия реализации программы

Обучение по программе реализовано в формате смешанного обучения, с применением активных технологий совместного обучения в электронной среде (синхронные и асинхронные занятия). Лекционный материал представляется в виде синхронных лекций, записей занятий, текстовых материалов, презентаций, размещаемых в электронном курсе. Данные материалы сопровождаются заданиями и дискуссиями в чатах дисциплин. Изучение теоретического материала (СРС) предполагается до и после синхронной части работы.

Материально-технические условия реализации программы

Синхронные занятия реализуются на базе инструментов видеоконференцсвязи и включают в себя лекционные и практические занятия. Для проведения синхронных занятий (вебинаров со спикерами) применяется программа видеоконференцсвязи. При проведении лекций, практических занятий, самостоятельной работы слушателей используется следующее оборудование: компьютер с наушниками или аудиокolonками, микрофоном и веб-камерой. Программное обеспечение (обновленное до последней версии): браузер Google Chrome, текстовый редактор, Microsoft Power Point.

Учебно-методическое и информационное обеспечение программы

Дисциплина может быть реализована как очно, так и заочно, в том числе, с применением дистанционных образовательных технологий. Она включает занятия лекционного типа, интерактивные формы обучения, практические занятия.

Содержание комплекта учебно-методических материалов

По данной дисциплине имеется электронный учебно-методический комплекс (УМК) в системе электронных курсов СФУ. УМК содержит: систему навигации по дисциплине (учебно-тематический план, интерактивный график работы по дисциплине, сведения о результатах обучения, чат для объявлений и вопросов преподавателю), текстовые материалы к лекциям, практические и тестовые задания, списки основной и дополнительной литературы. В электронном курсе реализована система обратной связи.

Литература

Основная литература

1. Собочинский И.Л. Проектирование информационных систем: учеб. пособие. – Красноярск: ИПК СФУ, 2007.
2. Шасанкар К., Кузнецов А. Zend Framework 2.0. Разработка веб-приложений. – СПб.: Питер, 2014.
3. Парамонова Т.Н. Маркетинг: учебник для вузов по направлению подготовки «Коммерция» и специальностям «Коммерция (торговое дело)», «Маркетинг», «Реклама». – М.: КНОРУС, 2016.
4. Горфинкель В.Я. Экономика фирмы: учебник для бакалавров. – М.: Юрайт, 2012.

4. Оценка качества освоения программы дисциплины (формы аттестации, оценочные и методические материалы)

Форма аттестации по дисциплине — зачет.

Оценка результатов обучения осуществляется следующим образом. Максимально за курс можно набрать 100%, из них:

- тесты самоконтроля к лекциям 40 %;
- практические задания составляют 60 %.

Зачет получают слушатели, набравшие не менее 50 % из 100 от общего прогресса по курсу.

Примеры тестов для контроля знаний

Пример тестового задания по типу «Выбор верного ответа»

1. Один из принципов сбора команды:
 - а) раздавайте опционы аккуратно;
 - б) поддерживайте рабочие отношения в команде.
 - в) не поручайте поиск людей другим;
 - г) не пытайтесь стимулировать деньгами;

2. Осуществление модификации возможно по модели:
- а) SCAMPER;
 - б) SCAMREP;
 - в) PERSCAM;
 - г) REPMASC.

Типовое практическое задание
Тема «Расчет ключевых метрик проекта»

Поставьте целевые показатели (ключевые метрики) проекта, используя прикрепленный шаблон.

Таблица 1. Ключевые метрики проекта

Целевой показатель	Значение целевого показателя сейчас	Целевое значение показателя через полгода после начала продаж	Целевое значение показателя через год после начала продаж

Критерии оценивания заданий

Баллы	1 балл	2 балла	3 балла	4 балла	5 баллов
Критерий	Реализован только вывод поля и персонажа	Реализовано только перемещение и выполнены предыдущие требования	Реализован поворот персонажа по направлению движения и выполнены предыдущие требования	Реализовано плавное перемещение персонажа и выполнены предыдущие требования	Реализован связанный с PlayerController класс Blueprint и выполнены предыдущие требования.

РАБОЧАЯ ПРОГРАММА СТАЖИРОВКИ

1. Аннотация

Основной задачей стажировки слушателей программы является закрепление в практической деятельности профессиональных компетенций, умений, навыков и знаний, полученных в ходе обучения, а также приобретение необходимых умений и практического опыта на конкретном рабочем месте.

Цель стажировки — приобретение слушателями программы практического опыта работы, а также освоение новых технологий, форм и методов организации труда непосредственно на рабочем месте.

Планируемые результаты:

По окончании стажировки слушатели будут способны составлять формализованные описания решений и разрабатывать алгоритмы решений поставленных задач в соответствии с требованиями технического задания, дизайнерского документа или других принятых в организации нормативных документов; применять навыки работы с движком Unity в комбинации с язык программирования C#; осуществлять проверку работоспособности программного обеспечения на основе разработанных тестовых наборов данных; использовать при разработке программного обеспечения существующие типовые решения и шаблоны проектирования, будут способны применять методы анализа предметной области, выявления проблем, составлять ценностного предложения, проводить анализ рынка, выявлять конкурентные преимущества проекта, создавать презентации для представления проекта.

2. Содержание

№, наименование темы	Содержание лекций (кол-во часов)	Наименование практических (семинарских занятий) (кол-во часов)	Виды СРС (кол-во часов)
Стажировка (16 часов)			
1. Общие вопросы (ознакомление с предприятием)		Ознакомление и изучение конкретной практической задачи (2 ч.)	
2. Практическая часть стажировки		Решение практической задачи (4 ч.) Интеграция собственного решения в общий проект (2 ч.)	
3. Подготовка отчетной документации			Составление отчета (4 ч.)

Содержание стажировки включает следующие этапы:

1. Ознакомление с нормативной базой, касающейся охраны труда и правил безопасной работы.

2. Знакомство с рабочим местом и должностными обязанностями, концептом общего тестового проекта.

3. Практическая деятельность, выполняемая под контролем руководителя стажировки. Обычно включает этапы:

3.1. Формирование отдельной практической задачи по общему проекту;

Содержание стажировки закрепляется индивидуальным планом прохождения стажировки (Приложение 1).

Продолжительность стажировки — 16 часов.

Стажировка носит индивидуальный или групповой характер и может предусматривать такие виды деятельности как:

- знакомство с предприятием, организационной структурой;
- изучение организации и технологии производства, работ;
- анализ производства;
- Знакомство с общим игровым проектом;
- работу с технической, нормативной и образовательной документацией;
- составление формализованных описаний решений поставленных задач;
- разработку алгоритмов решений поставленных задач в соответствии с требованиями технического задания;
- Написание программного кода на языке C#;
- Тестирование программного кода;
- Интеграцию программных модулей;
- Представление проекта.

3. Условия реализации программы стажировки

Организационные и педагогические условия реализации программы

Обучение по программе стажировки реализовано в формате смешанного обучения, с применением активных технологий совместного обучения в электронной среде (синхронные и асинхронные занятия). Материал практических занятий представляется в виде синхронных занятий, презентаций, размещаемых в электронном курсе. Данные материалы сопровождаются заданиями и дискуссиями в чатах дисциплин. Изучение теоретического материала (СРС) предполагается до и после синхронной части работы.

Стажировка проводится под руководством назначенного руководителя из числа профессорско-преподавательского состава Университета, а также руководителя из состава организации, структурных подразделениях организации, материально-техническое обеспечение которой соответствует профилю программы.

Учебно-методическое и информационное обеспечение

По данному модулю используется электронный УМК. УМК предполагает использование разных типов материалов, сопровождающих учебный процесс, включая информационные, обучающие и контролирующие. На платформе электронных курсов размещаются задания, приводится перечень необходимых для изучения материалов. Обучающиеся могут на протяжении прохождения стажировки обращаться к теоретической базе знаний.

4. Оценка качества освоения программы стажировки (формы аттестации, оценочные и методические материалы)

В качестве подтверждения прохождения стажировки на базе предприятий, организаций, учреждений, для зачета результатов обучения слушателями предъявляется дневник прохождения стажировки (Приложение 2) (*отчет в виде дневника прохождения практики*).

Программу составили:

ассистент кафедры
Информационных систем ИКИТ СФУ,

 А.С. Зайцев

ассистент кафедры
Информационных систем ИКИТ СФУ,

 И.А. Чудакин

ассистент кафедры
Информационных систем ИКИТ СФУ,

 Е.А. Халтурин

Руководитель программы:

канд. пед. наук, доцент,
Заведующий кафедрой
Прикладной информатики ИКИТ СФУ

 П.П. Дьячук

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Наименование образовательной организации

Индивидуальный план слушателя, направляемого на стажировку

Фамилия, имя, отчество _____

Место работы и должность/статус _____

Название предприятия (организации), где проводится стажировка

Город _____

Цель стажировки _____

Срок стажировки с «___» _____ 2023 г. по «___» _____ 202 г.

Приказ по вузу от «___» _____ 202 г. № _____

План стажировки

№ п.п.	Перечень разрабатываемых (изучаемых) вопросов, виды работ	Количество часов	Форма отчета
1.			Дневник стажировки
2.			
3.	Заполнение дневника стажировки		

СОГЛАСОВАНО

(должность ответственного)

(подпись)

(расшифровка подписи) лица,
направляющего на стажировку)

Наименование стажировочной площадки

УТВЕРЖДАЮ

Руководитель стажировочной площадки

_____ ФИО

«_____» _____ 2022 г.

М.П.

**ДНЕВНИК
прохождения стажировки**

_____,
(фамилия, имя, отчество специалиста (стажера),
проходящего обучение в рамках дополнительной профессиональной программе
переподготовки «Разработка мобильных игр на Unity»

Цель стажировки:

Руководители стажировки (от организации): _____
(должность) (ФИО)

1. Дневник

Дата	Выполняемая работа	Вопросы для консультантов и руководителей стажировки

2. Краткий отчет о стажировке

Дата

Подпись стажера

3. Заключение руководителя стажировки от принимающей организации

Руководитель стажировки

(подпись)

(расшифровка подписи)

С заключением руководителя стажировки ознакомлен _____
(подпись стажера)