Министерство образования Края
Филиал АНО ДТ «Красноярский «Кванториум» в г. Норильске
«Центр цифрового образования детей IT-Куб г. Норильск»

СОГЛАСОВАНО:
Председатель методического совета
_____Т.С.Новгородцева
протокол № 11
от «29» мая 2026 г.

УТВЕРЖДАЮ:
Директор филиала
_____Е. А. Дыптан
Приказ № 02-02-92
от «01» 06 2026 г.

ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА НА 2026– 2027 уч. год
технической направленности

«Основы Unity»

Форма реализации программы – очная
Срок реализации – 1 год
Возраст обучающихся – 12-14 лет

Составитель программы:
Александров Михаил Максимович
педагог дополнительного образования

г. Норильск, 2026 г.

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Дополнительная общеобразовательная общеразвивающая программа «Основы Unity» имеет техническую направленность и разработана в соответствии с основными нормативно-правовыми документами: Федеральным Законом «Об образовании» от 29.12.2012 г. № 273-ФЗ; Порядком организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам от 09.11.2018 г. № 196; Целевой моделью развития региональных систем дополнительного образования детей от 03.09.2019 г. № 467; Концепцией развития дополнительного образования детей до 2030 года утвержденной распоряжением Правительства Российской Федерации от 31.03.2022 №678-р; Санитарно-эпидемиологическими требованиями к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи от 28.09.2020 г. № 28.

Уровень программы – стартовый.

С каждым годом разработка игр становится популярнее, доступнее и привлекательнее для подрастающих специалистов. В связи с этим, появилась необходимость профориентации и более доступной информации для учащихся школ. Освоив навыки, применяемые при разработке игр и разобравшись в среде программирования Unity, слушатели смогут решить хотят ли они дальше развиваться в этой области и в какой роли.

Программа нацелена на слушателей, умеющих пользоваться компьютером и сетью интернет, владение навыками и знаниями в программировании, алгоритмике, разработке игр не обязательны. Программа позволит расширить компетенции слушателей, проводя их через увлекательную и многогранную сферу разработки компьютерных игр. Сферу, в которой они найдут применении множеству различных навыков и знаний из жизненного опыта, науки и творчества, поскольку множество вещей вокруг могут быть игрой или быть представлены в ней.

Обучение основам Unity способствуют формированию современного научного мировоззрения, развитию интеллектуальных способностей и познавательных интересов обучающихся, развивает способности к самообразованию, умению ставить задачи, планировать их выполнение, умению разбивать решение задачи на подзадачи, способность работать с информацией: находить, оценивать и использовать информацию из различных источников. Обучение по данной программе дает обучающимся возможность попробовать силы в реализации собственных идей и желаний. Создание игр— это не просто знание какого-то языка, инструмента или технологии, это способность оживить свои идеи, или найти решения какой-то проблеме создав необходимую для этого симуляцию. После прохождения данной программы слушателю будет проще определиться с дальнейшим направлением профессионального роста и набору компетенций.

1.1 НОВИЗНА ПРОГРАММЫ

Новизна программы заключается в том, что данный курс не только обучает детей программированию и созданию собственных компьютерных игр, а также формирует у обучающихся представления о многогранности компьютерных игр. Таких как технико-педагогические (обучающие и управляющие, диагностирующие, моделирующие, экспертные, диалоговые, консультирующие, расчетно-логические), стратегии, аркады, квесты и др.

Предполагается большое количество практики создания игр в среде Unity.

Программа рассчитана на следующие этапы обучения:

1. Знакомство с разработкой игр.
2. Освоение методов разработки игр в среде Unity.
3. Освоение базовых принципов разработки, алгоритмики и геймдизайна.
4. Практика создания игр по примерам и собственным идеям слушателей при помощи наставника.

1.2 АКТУАЛЬНОСТЬ ПРОГРАММЫ

Актуальность программы обусловлена большим запросом со стороны юный любителей компьютерных игр. Геймдев считается одним из самых перспективных направлений в области информационных технологий. Изучение данной программы не только расширит кругозор. Она поможет ребенку понять, кем он хочет стать – профессиональным геймером, или все-таки тем, кто будет разрабатывать компьютерные игры.

Unity – среда разработки кроссплатформенных интерактивных приложений и игр. Создаваемая на движке Unity3D игра уже сама по себе призвана стать шедевром, так как она будет удивлять игроков своими непревзойденными технологическими возможностями. В использовании Unity 3D нет сложностей написания кода, так как он поддерживает скриптовые языки разработки, нет необходимости пользоваться большим количеством сторонних инструментов, в связи с тем, что в среде Unity 3D интегрировано большое количество мощных вспомогательных средств, связанных с проектированием и дизайном. Работать в среде Unity просто и удобно, элементы интерфейса наглядны и интуитивно понятны. Перечисленное и определяет актуальность данной дополнительной общеразвивающей образовательной программы.

1.3 ПЕДАГОГИЧЕСКАЯ ЦЕЛЕСООБРАЗНОСТЬ

Педагогическая целесообразность программы заключается в ее метапредметности. Знания, умения и навыки, полученные в ходе освоения программы, помогут обучающемуся оптимально использовать информационные технологии и навыки

проектной деятельности для решения различных задач. Практическая направленность программы может способствовать профессиональному самоопределению обучающихся.

Разработка игр — это не тривиальная задача и поэтому курс построен таким образом, чтобы помочь учащимся заняться разработкой, найти ответы на вопросы, с которыми им приходится сталкиваться в повседневной жизни и понять в каком направлении им хотелось бы развиваться дальше. Многие люди желают стать разработчиками игр, не зная о многообразии сфер приложения и огромном количестве направлений в этой области. Применение слушателями на практике теоретических знаний, полученных в школе на математике и физике, ведет к более глубокому пониманию основ, закрепляет полученные навыки, формируя понимание того, для чего в школе изучается теоретический материал на уроках, который не всегда очевидным образом применяется в реальной жизни и в реальных задачах.

Изучая основы Unity, слушатели получают глубокое понимание принципов работы компьютера, программ, процесса создания игр, организации ввода, вывода и хранения информации, познают азы направления разработки игр.

Все задачи, которые необходимо решить детям, будут даваться в игровой форме. В конце курса, в качестве итоговой аттестации, обучающиеся напишут свою игру, используя все знания, которые они получили во время прохождения курса.

1.4 ЦЕЛЬ ПРОГРАММЫ

Цель программы — сформировать навыки работы с прикладным программным обеспечением Unity для создания собственных проектов.

Задачи:

- сформировать у обучающихся базовые представления о сфере разработки игр;
- сформировать представление о профессии разработчика игр;
- сформировать навыки составления основных алгоритмических конструкций (линейных, разветвляющихся и циклических);
- сформировать навык использования игровых движков для создания игр;
- познакомить с понятием проекта и алгоритмом его разработки;
- развивать самостоятельность и творческий подход к решению задач с использованием программного обеспечения Unity и прикладных программ по работе с графикой, звуком;
- приобрести навыки поиска информации в сети Интернет, научиться анализировать найденную информацию на соответствие запросу и использовать её при решении задач;
- расширить кругозор обучающихся в области разработки программных решений;

- приобрести навыки самостоятельной организации своей деятельности; сформировать основы для саморазвития и самовоспитания;
- развить готовность и способность к самостоятельной, творческой деятельности, к образованию, в том числе самообразованию; готовность к осознанному выбору будущей профессии.

1.5 ВОЗРАСТ ОБУЧАЮЩИХСЯ, УЧАСТВУЮЩИХ В ПРОГРАММЕ

Программа «Основы Unity» рассчитана на обучающихся 12-14 лет.

1.6 УСЛОВИЯ ВХОЖДЕНИЯ В ПРОГРАММУ:

Набор на Программу осуществляется в соответствии с Положением о наборе в АНО ДТ «Красноярский «Кванториум».

1.7 СРОК РЕАЛИЗАЦИИ ПРОГРАММЫ:

Программа рассчитана на 1 год обучения. Годовая нагрузка на обучающегося составляет 144 часа.

1.8 РЕЖИМ ЗАНЯТИЙ, ФОРМЫ И МЕТОДЫ ОБУЧЕНИЯ

Учебные занятия проходят по очной форме обучения. Режим занятий – 2 раза в неделю по 2 академических часа (1 академический час — 40 минут) с обязательным перерывом.

Форма проведения занятий: фронтальная, групповая, индивидуальная.

Формы проведения учебных занятий соответствуют содержанию программы и предусматривают: лекции, консультации, практические занятия, круглые столы, мастер-классы, защита проектов, соревнования и конкурсы.

1.9 ОЖИДАЕМЫЕ ОБРАЗОВАТЕЛЬНЫЕ РЕЗУЛЬТАТЫ И ЭФФЕКТЫ, СПОСОБЫ ПРЕДЪЯВЛЕНИЯ И ОТСЛЕЖИВАНИЯ РЕЗУЛЬТАТОВ

По результатам обучения, слушатели овладевают основами работы в среде Unity, имеют представление о разработке игр.

Учащиеся будут знать:

- возможности разработки;
- основные отличия визуального программирования от традиционного;
- основные приемы создания программ-приложений;
- требования к написанию и оформлению программ-приложений;
- основные принципы работы компьютера и выполнения программы;
- основные алгоритмические конструкции.

Учащиеся будут уметь:

- работать с основными инструментами и пользоваться средой Unity для разработки игр;
- создавать дизайн уровня;
- создавать собственного персонажа;
- работать с физикой в Unity;
- создавать анимацию;
- разрабатывать законченные компьютерные игры;
- создавать и искать игровые ресурсы;
- разрабатывать алгоритм задачи;
- работать в сети интернет для поиска информации.

Личностные результаты:

- учащиеся приобретут навыки самостоятельной организации своей деятельности; будут сформированы основы для саморазвития и самовоспитания;
- у слушателей сформируется готовность и способность к самостоятельной, творческой деятельности, к образованию, в том числе самообразованию; готовность к осознанному выбору будущей профессии.

Метапредметные результаты:

- учащиеся приобретут навыки поиска информации в сети Интернет, научатся анализировать найденную информацию на соответствие запросу и использовать её при решении задач.

Предметные результаты:

- сформированы базовые представления о сфере разработки игр;
- сформировано представление о профессии разработчика игр;
- сформированы навыки составления основных алгоритмических конструкций (линейных, разветвляющихся и циклических);
- сформирован навык использования игровых движков для создания игр;
- усвоено понятие проекта и алгоритма его разработки;
- развиты самостоятельность и творческий подход к решению задач с использованием программного обеспечения Unity и прикладных программ по работе с графикой, звуком.

Механизм оценки результативности

Контроль качества образования осуществляется в форме опроса, выполнения практических и проектных работ.

По итогам каждого раздела программы проводится текущий контроль в форме небольшой проектной работы.

Итоговая аттестация осуществляется по результатам теоретического тестирования и успешно защищенном проекте (созданной компьютерной игры).

2. УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН

№	Наименование разделов	Всего часов	Теория	Практика	Форма контроля
1	**Знакомство с направлением обучения**	**4**	**3**	**1**	
1.1	Вводный урок. Правила работы и техника безопасности при работе на компьютере	2	1	1	
1.2	Содержание курса. Разработка игр	2	2	0	
2	**Геймдизайн**	**24**	**11**	**13**	
2.1	Что такое игры	4	2	2	
2.2	Разработка игр. Геймплей	4	2	2	
2.3	Повествование в играх. Сюжеты. Диалоги	4	2	2	
2.4	Инструменты разработки. Дополнительное ПО	2	1	1	
2.5	Создание собственных миров	4	2	2	
2.6	Логика и алгоритмы	4	2	2	
2.7	Текущий контроль по разделу «Геймдизайн». Закрепление материала	2	0	2	Опрос по разделу, небольшой проект
3	**Unity**	**34**	**11**	**23**	
3.1	Введение в Unity	2	1	1	
3.2	Рабочее поле. Инструменты. Дополнительные модули	4	2	2	
3.3	Прототипирование. Игровые объекты	8	2	6	
3.4	Создание сцен. Скрипты. Переходы	4	2	2	
3.5	Игровые ресурсы. Материалы. Шейдеры	8	2	6	
3.6	Физика	6	2	4	

3.7	Текущий контроль по разделу «Unity». Закрепление материала	2	0	2	Опрос по разделу, небольшой проект
4	**Практика**	**82**	**22**	**60**	
4.1	Скриптинг. Ввод. Вывод. Игра «Викторина»	4	2	2	
4.2	Условия. Генерация. UI. Игра «Рефлекс»	6	2	4	
4.3	Передвижение. Столкновения. Игра «Платформер»	10	2	8	
4.4	Игра «Тир»	4	2	2	
4.5	Роботы. Взаимодействие объектов. Игра «Гонки»	6	2	4	
4.6	Работа со звуком. Машины состояний. Алгоритмы. Игра «Прятки»	8	2	6	
4.7	Обучение. Взаимодействие с игроком. Сохранение. Конец игры. Игра «Подземелье»	10	2	8	
4.8	Постобрабокта. Эффекты. Трехмерное пространство	4	2	2	
4.9	Итоговая аттестация. Создание собственной игры	30	6	24	Защита проекта, теоретическое тестирование
	ИТОГ	**144**	**47**	**97**	

3. СОДЕРЖАНИЕ ПРОГРАММЫ

Раздел 1. Знакомство с направлением обучения (4 часа).

Тема 1.1. Вводный урок. Правила работы и техника безопасности при работе на компьютере (2 часа).

Теоретическая работа: вводная лекция, включающая в себя правила техники безопасности и общего распорядка.

Тема 1.2. Содержание курса. Разработка игр (2 часа).

Теоретическая работа: Пояснение по содержанию программы. Планируемые результаты и компетенции.

Раздел 2. Геймдизайн (24 часа).

Тема 2.1. Что такое игры? (4 часа)

Теоретическая работа: Игры как средство выразительности. Развлекательная составляющая. История компьютерных игр.

Практическая работа: Высказывание своего мнения по поводу игр. Обсуждение любимых проектов.

Тема 2.2. Разработка игр. Геймплей (4 часа).

Теоретическая работа: Основы разработки игр. Механики. Понятие геймплея.

Практическая работа: Придумывание собственных игр и продумывание механик.

Тема 2.3. Повествование в играх. Сюжеты. Диалоги (4 часа).

Теоретическая работа: Виды повествования в играх. Сюжет и его влияние. Варианты диалогов и взаимодействия с игроком и внутри игры.

Практическая работа: Создание собственных историй, сюжетов и диалогов.

Тема 2.4. Инструменты разработки. Дополнительное ПО (2 часа).

Теоретическая работа: Существующие инструменты для разработки игр. Используемое программное обеспечение.

Практическая работа: Сравнение программного обеспечения. Скачивание, установка, настройка.

Тема 2.5. Создание собственных миров (4 часа).

Теоретическая работа: как создавать собственные миры. Логика. Правила.

Практическая работа: Создание собственных миров на бумаге

Тема 2.6. Логика и алгоритмы (4 часа).

Теоретическая работа: Основные понятия логики и алгоритмизации.

Практическая работа: Составление простых алгоритмов.

Тема 2.7. Текущий контроль (2 часа).

Практическая работа: Опрос по разделу. Закрепление материала.

Раздел 3. Unity (34 часа).

Тема 3.1. Введение в Unity (2 часа).

Теоретическая работа: Что такое Unity. Применение. Возможности. Примеры проектов.

Практическая работа: Оценка существующих проектов. Мозговой штурм по идеям для проектов.

Тема 3.2. Рабочее поле. Инструменты. Дополнительные модули (4 часа).

Теоретическая работа: Настройка рабочего поля. Встроенные инструменты редактора Unity. Горячие клавиши. Загружаемые модули.

Практическая работа: Поломка рабочего поля. Эксперименты с инструментами. Загрузка и настройка модулей.

Тема 3.3. Прототипирование. Игровые объекты (8 часов).

Теоретическая работа: Возможные игровые объекты в Unity. Взаимодействие с ними. Как создавать прототип игр.

Практическая работа: Эксперименты с игровыми объектами. Начало прототипа игры.

Тема 3.4. Создание сцен. Скрипты. Переходы (4 часа).

Теоретическая работа: Сцены в Unity. Создание скриптов. Переходы между сценами.

Практическая работа: Создание первой сцены. Осмотр скриптов. Переходы между уровнями.

Тема 3.5. Игровые ресурсы. Материалы. Шейдеры (8 часов).

Теоретическая работа: Создание и загрузка игровых ресурсов. Обработка. Настройка материалов и шейдеров.

Практическая работа: Настройка игровых ресурсов. Размещение и настройка материалов.

Тема 3.6. Физика (6 часов).

Теоретическая работа: Физическая симуляция в Unity. Работа с физикой. Проверка столкновений.

Практическая работа: Взаимодействие с физическими объектами в Unity.

Тема 3.7. Текущий контроль (2 часа)

Практическая работа: Опрос по разделу. Закрепление материала

Раздел 4. Практика (82 часа).

Тема 4.1. Скриптинг. Ввод. Вывод. Игра «Викторина» (4 часа).

Теоретическая работа: Методы ввода и вывода информации. Управление. Принцип игры в викторину. Создание скриптов.

Практическая работа: Закрепление материала во время создания игрового проекта.

Тема 4.2. Условия. Генерация. UI. Игра «Рефлекс» (6 часов).

Теоретическая работа: Изменение условий игры. Генерация объектов.

Практическая работа: Закрепление материала во время создания игрового проекта.

Тема 4.3. Передвижение. Столкновения. Игра «Платформер» (10 часов).

Теоретическая работа: Понимание передвижение в игровом пространстве. Автоматическая обработка столкновений.

Практическая работа: Закрепление материала во время создания игрового проекта.

Тема 4.4. Игра «Тир» (4 часа).

Теоретическая работа: Pooling объектов. Raycast.

Практическая работа: Закрепление материала во время создания игрового проекта.

Тема 4.5. Роботы. Взаимодействие объектов. Игра «Гонки» (6 часов).

Теоретическая работа: Создание простого ИИ. Навигационные агенты.

Практическая работа: Закрепление материала во время создания игрового проекта.

Тема 4.6. Работа со звуком. Машины состояний. Алгоритмы. Игра «Прятки» (8 часов).

Теоретическая работа: Звук в Unity. Производство звуков и музыки своими руками. Обработка состояний игровых объектов. Внедрение алгоритмов.

Практическая работа: Закрепление материала во время создания игрового проекта.

Тема 4.7. Обучение. Взаимодействие с игроком. Сохранение. Конец игры. Игра «Подземелье» (10 часов).

Теоретическая работа: Обучение в играх. Диалог с игроком. Варианты сохранения информации. Условия окончания игры. Поощрение игроков. Внедрение элемента случайности. Создание RPG элементов и их внедрение в игру.

Практическая работа: Закрепление материала во время создания игрового проекта.

Тема 4.8. Постобработка. Эффекты. Трехмерное пространство (4 часа).

Теоретическая работа: Применение визуальных эффектов. Настройка визуальных эффектов и системы частиц. Использование трехмерного пространства.

Практическая работа: Добавление эффектов и улучшения визуальной составляющей своих проектов.

Тема 4.9. Итоговая аттестация (30 часов).

Практическая работа: Создание собственной игры. Защита творческого проекта.

4. МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА, НЕОБХОДИМАЯ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ПРОГРАММЕ

Помещения для занятий и самостоятельной работы обучающихся должны быть оснащены персональными компьютерами с установленными программными пакетами актуальных версий Unity, Microsoft Office , с возможностью подключения к сети Интернет, доской с маркером или интерактивной доской, мультимедийным проектором.

5. РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

Основная литература:

1. Джозеф Хокинг, Unity в действии. Мультиплатформенная разработка на C# / Питер, 2019
2. Джереми Гибсон Бонд, Unity и C#. Геймдев от идеи до реализации / Питер, 2020.
3. Джесси Шелл, Геймдизайн / Альпина Паблишер, 2019.
4. Рэф Костер, Разработка игр и теория развлечений / ДМК, 2018.
5. Джейсон Шреер, Кровь, пот и пиксели / Эксмо, 2018.

Электронные ресурсы:

1. google.com/
2. habr.com/
3. unity3d.com/

Перечень программного обеспечения:

1. notion.so/
2. unity3d.com/

6. СВЕДЕНИЯ О СОСТАВИТЕЛЯХ ДООП

А. М. Петров – доцент кафедры информационных систем и технологий, педагог высшей категории, педагог дополнительного образования по направлению «Unity» и «VR/AR». Образование высшее. Ученая степень кандидата технических наук. Магистр психологии.

ТЕМЫ ИНДИВИДУАЛЬНЫХ ПРОЕКТОВ:

1. Платформер.
2. Симулятор шахтера.
3. Гоночный симулятор.
4. Игра в прятки.
5. Выживание.
6. Рыбалка.
7. Тир.
8. Постройка города.

Тест №1.

1. Какие объекты добавляются по умолчанию? (имеется в виду — какие объекты добавлены сразу же на сцену при создании пустого 3D проекта)

- лишь объект Directional light
- лишь объект Main Camera
- объекты по умолчанию не добавляются
- объекты Main Camera, Directional light и один пустой объект
- объекты Main Camera и Directional light

2. За что отвечает вкладка Hierarchy?

- В ней находятся все материалы к проекту
- В ней располагаются свойства к объектам
- В ней располагаются объекты на сцене
- В нее выводятся различные ошибки и надписи в ходе игры
- В ней отображается игровая сцена

3. Какой компонент есть у каждого объекта?

- Rigidbody
- Mesh Renderer
- Collider
- Transform
- Light

4. Какая вкладка отвечает за отображение сообщений?
    - вкладке Hierarchy
    - вкладка Project
    - вкладке Console
    - вкладка Inspector
5. Как называется официальный магазин от Unity?
    - Packages Store
    - Projects Store
    - Unity Store
    - Asset Store
    - Plugins Stor
6. Можно ли работать в бесплатной версии Unity?
    - Да, можно без ограничений
    - Можно, но лишь до того, как ваш годовой доход не превысил 100 000 $ за год
    - Нет, всегда необходимо докупать платную версию
7. Что отображается на вкладке Project?
    - В ней располагаются все объекты на сцене
    - В нее выводятся различные ошибки и надписи в ходе игры
    - В ней находятся все материалы к проекту
    - В ней располагаются свойства к объектам
    - В ней отображается игровая сцена
8. Для чего нужна вкладка Navigator?
    - вкладка позволяет создать агентов
    - она позволяет указать объекты, по которым можно ходить
    - вкладка позволяет создать агента и поверхность, по которой сможет передвигаться агент
    - она служит для создания игровых объектов на сцене
    - в ней создается искусственный интеллект для игры

9. Можно ли вращать и передвигать камеру?
   - Нет, нельзя
   - Можно, только если камера не основная
   - Можно всегда
10. За что отвечает вкладка Lighting?
   - она используется для настройки лишь света в игре
   - она позволяет указать лишь skybox для основной камеры
   - в ней можно настроить свойства Direction Light
   - вкладка отвечает за работу со светом на сцене, а также позволяет установить туман

Тест №2

https://wordwall.net/ru/resource/56779419/%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0/%D1%82%D0%B5%D1%81%D1%82-%D0%BF%D0%BE-unity по ссылке

Тест №3 (итоговый)

1. Unity — это...
   - название компьютерной игры
   - название компании, занимающейся разработкой игр
   - среда разработки приложений и компьютерных игр
   - логотип, которым отмечают качественные игры
2. Чтобы начать работать с Unity необходимо:
   - быть опытным программистом со стажем
   - скачать и установить программу на свой компьютер и пользоваться предоставленными методическими пособиями
   - получить официальное разрешение на разработку
   - купить лицензию от Unity Technologies
3. Чтобы установить Unity на компьютер необходимо:
   - около 10 ГБ свободного места на диске С
   - не менее 25 ГБ свободного места на любом диске
   - не менее 50 ГБ свободного места на диске D
   - на всех дисках - по 10 ГБ свободого места
4. Чтобы создавать приложения и игры на Unity обязательно нужно:

- уметь создавать графику для приложений и компьютерных игр
- уметь писать скрипты на языке C#
- просто придумать идею хорошей программы или игры
- освоить интерфейс программы и можно приступать к разработке

5. Unity позволяет создавать:
   - любые игры разного уровня сложности
   - любые приложения, игры, а также графику для кинематографических и анимационных проектов
   - игры для запуска на iOS и Android
   - игры для телефонов, ПК и серверные игры
6. Графику для приложений и игр на Unity...
   - рекомендуется всегда рисовать самостоятельно
   - рекомендуется всегда добавлять из библиотеки Unity Asset Store, чтобы не было ошибок
   - можно использовать собственную или из наборов Asset Store, но с других сторонних сайтов брать ассеты с графикой запрещено
   - можно использовать любую: собственную, из наборов Asset Store или с других сайтов, содержащих ассеты с графикой
7. Освоить разработку на Unity...
   - очень сложная задача, особенно для начинающих
   - очень простая задача, особенно для начинающих
   - простая задача, если обучатся систематически, с помощью книг, уроков или видеокурсов
   - можно только с помощью персонального преподавателя
8. Зачем нужна вкладка Project?
   - С помощью вкладки Project можно управлять файлами проекта.
   - С помощью вкладки Project можно передвигать объекты на сцене.
   - С помощью вкладки Project можно менять параметры компонентов.
   - С помощью вкладки Project можно запустить игру.
9. Зачем нужна вкладка Hierarchy?
   - Вкладка Hierarchy содержит список всех компонентов выбранного GameObject'a.
   - Вкладка Hierarchy содержит список всех файлов выбранного проекта.
   - Вкладка Hierarchy содержит список всех GameObject'ов на сцене.

- Вкладка Hierarchy содержит список всех параметров выбранного материала.

10. В чем разница между Scene View и Game View? Выберите наиболее верный вариант.

- В Game View интерактивный вид на сцену, которою вы создаете и позволяет выделять, передвигать объекты и т.д. В Scene View отображается вид одной из камер, установленных на вашей сцене.
- В Scene View интерактивный вид на сцену, которою вы создаете и позволяет выделять, передвигать объекты и т.д. В Game View отображается вид одной из камер, установленных на вашей сцене.
- В Scene View вы можете менять параметры компонентов. В Game View вы можете поменять иерархию объектов.
- В Game View вы можете добавить новые файлы в проект. В Scene View вы можете поменять название GameObject'a.

11. Что отображается в панеле Inspector'a?

- В Inspector'e отображается список файлов, добавленных в проект.
- В Inspector'e отображается список префабов.
- В Inspector'e отображается список GameObject'ов на сцене.
- В Inspector'e отображается информация о выбранном GameObject'e, включая все прикрепленные компоненты и их параметры.

12. Что такое GameObject? Выберите наиболее верный вариант.

- GameObject это базовый объект на всех сценах Unity, позволяющий менять цвет материала.
- GameObject это второстепенный объект на всех сценах Unity, содержит название выбранного объекта.
- GameObject это базовый объект на всех сценах Unity, является контейнером для компонентов.
- GameObject это второстепенный объект на всех сценах Unity, содержит координаты и поворот объекта.

13.Что такое компонент?

- Компонент — это базовый геометрический объект Unity. Куб, сфера и т.д.
- Компонент это — базовый класс для всего, что прикрепляется к GameObject'y. Содержит элементы функционала Unity или скрипт.
- Компонент — это совокупность всех GameObject'ов, расположенных на сцене.

- Компонент — это базовый класс для всех спец. эффектов в Unity. Освещение, материал и т.д.

14. Что такое Transform?
    - Компонент Transform определяет Mass (массу), Velocity (скорость), Drag (сопротивление) объектов в сцене. У каждого GameObject’а есть Transform.
    - Компонент Transform определяет Position (положение), Rotation (вращение), и Scale (масштаб) каждого объекта в сцене. У каждого GameObject’а есть Transform.
    - Компонент Transform определяет Color (цвет), Texture (текстуру), Shader (шейдер) объектов в сцене. Не у всех GameObject'ов есть Transform.
    - Компонент Transform определяет Color (цвет), Mode (режим), Intencity (интенсивность) освещения объектов в сцене. У каждого GameObject’а есть Transform.
15. Из чего состоят 3D модели?
    - 3D модели состоят из точек в пространстве и треугольников, соединяющие эти точки.
    - 3D модели состоят из файла текстуры и поверхности.
    - 3D модели состоят из набора параметров, определяющих физическое взаимодействие с физическим движком.
    - 3D модели состоят из точек в пространстве и линий, соединяющих их.
16. Что такое Mesh в Unity3D?
    - Mesh это координатная сетка в Scene View, позволяющая точно расположить объекты на сцене.
    - Mesh это это файл изображения или фильма, который "натягивается" на 3D модель или отображается плоским на экране.
    - Mesh это любой GameObject, отображающийся в Game View.
    - Mesh это внутренний класс Unity, позволяющий создавать и изменять 3D модели с помощью скриптов.
17. Что такое Collider?
    - Collider это компонент, определяющий массу GameObject'a.
    - Collider это файл изображения или фильма, который "натягивается" на 3D модель или отображается плоским на экране.
    - Collider это компонент, определяющий физическую форму объекта, для взаимодействия с физикой игры.

- Collider это компонент, определяющий визуальную форму объекта, для взаимодействия с Renderer'ом игры.

18. Что такое Текстура?
    - Текстура — это любая 3D модель, имеющая цвет.
    - Текстура — это файл изображения или фильма, который "натягивается" на 3D модель или отображается плоским на экране.
    - Текстура — это невидимые границы сцены, не позволяющие игроку покинуть сцену.
    - Текстура — это параметры координат, поворота, масштаба GameObject'a.
19. Что такое Material?
    - Material это элемент Unity, определяет, как расположен GameObject. Совокупность параметров координат, поворота, масштаба.
    - Material это элемент Unity, определяет, как GameObject взаимодействует с физикой. Совокупность параметров массы, скорости, сопротивления.
    - Material это элемент Unity, определяет, как отображается поверхность. Совокупность параметров, текстур и Shader'a.
    - Material это элемент Unity, определяет, как работает GameObject. Совокупность компонентов.
20. Что такое Prefab?
    - Prefab это любой GameObject, сохраненный в файлах проекта.
    - Prefab это любой файл, сохраненный в файлах проекта.
    - Prefab это любой компонент, в котором заранее прописаны параметры.
    - Prefab это любой объект, отображаемый на сцене.
21. Что такое Родительский объект (Parent)?
    - Родительским объектом является GameObject, который подсоединен к другому GameObject.
    - Родительским объектом является компонент, подсоединенный к GameObject'y.
    - Родительским объектом является GameObject, к которому подсоединен хотя бы один компонент.
    - Родительским объектом является GameObject, к которому подсоединен другой GameObject.
22. В чем разница между локальными (Local) и глобальными (World) координатами?

- Глобальные координаты отсчитываются относительно родительского GameObject'a, а локальные координаты отсчитываются относительно центра сцены.
- Локальные координаты отсчитываются относительно родительского GameObject'a, а глобальные координаты отсчитываются относительно центра сцены.
- Глобальные координаты одинаковы для всех сцен, а локальные уникальны для конкретной сцены.
- Глобальные координаты сохраняются в Prefab'e, а локальные нет.

23. Что такое шейдер (Shader)?
    - Shader это маленький скрипт, который содержит математические формулы и алгоритмы для расчёта цвета поверхности или отдельного пикселя.
    - Shader это компонент Unity, содержит геометрию 3D модели. Работает в соответствии с параметрами компонента Transform.
    - Shader это набор параметров и настроек освещения, определяет конечный цвет поверхности.
    - Shader это дополнительная система для обработки любого 3D изображения, позволяет улучшить качество графики. Unity может отображать низкокачественную графику без Shader'a.
24. Что такое карта нормалей (Normal Map)?
    - Карта нормалей — это тип шейдера в Unity, позволяет отображать мелкие детали на поверхности 3D модели.
    - Карта нормалей — это тип текстуры, позволяет отображать мелкие детали на поверхности 3D модели.
    - Карта нормалей — это мелкие детали 3D модели.
    - Карта нормалей — часть 3D модели, позволяет поверхности 3D модели взаимодействовать с освещением.
25. Что такое терэйн (Terrain)?
    - Terrain это система, встроенная в редактор Unity, позволяющая создавать ландшафт.
    - Terrain это любой ландшафт, созданный в редакторе Unity.
    - Terrain это ландшафт, созданный в сторонней программе, а затем импортированный в Unity.
    - Terrain это любой GameObject, на который присвоена текстура земли.

26. Что такое Asset Store?
- Это сайт, содержащий множество дополнительных ассетов, систем и компонентов, созданных разработчиками Unity.
- Это сайт с документацией для Unity. Содержит множество полезных справок и описание функционала Unity.
- Это сообщество пользователей Unity, которое бесплатно распространяет готовые компоненты и системы для Unity. Встроен в редактор Unity.
- Это магазин всевозможных дополнительных компонентов, систем и ассетов, созданных сторонними разработчиками и разработчиками Unity. Встроен в редактор Unity.

27. Как запустить приложение в редакторе Unity?
- На клавиатуре одновременно нажать Alt и F4.
- Сверху-слева нажать File -> Build Settings -> Build and Run
- На клавиатуре одновременно нажать Contol и английскую букву S.
- Нажать кнопку Play сверху-посередине интерфейса Unity.

28. Как собрать приложение или игру для Windows?
- File -> Build Settings -> Выбрать платформу WebGL -> Build
- File -> Build Settings -> Выбрать платформу Standalone -> Выбрать target platform Windows -> Build
- File -> Build Settings -> Build and Run
- File -> Build Settings -> Выбрать платформу Standalone -> Выбрать target platform Ubuntu -> Build

29. Что такое "компиляция"?
- Компиляция — это процесс, в котором компилятор переводит текст скрипта в машинные команды.
- Компиляция — это процесс сохранения сцены в Unity.
- Компиляция — это процесс сборки ассет пакаджа (Asset Package).
- Компиляция — это процесс добавления новых ассетов (Asset) в Unity.

30. Что такое ".exe файл"?
- .exe file это файл, который содержит геометрию 3D модели в Unity.
- .exe file это файл, который содержит исполняемый скрипт для оболочки Windows.
- .exe file это файл, который содержит двоичную информацию о текстуре.

- .exe file это файл, который содержит выполняемую программу для Windows.

31. Что такое Rigidbody?
    - Компонент Rigidbody определяет Color (цвет), Texture (текстуру), Shader (шейдер) объектов в сцене. Не у всех GameObject'ов есть Transform.
    - Компонент Rigidbody определяет Mass (массу), Velocity (скорость), Drag (сопротивление) и физическое взаимодействие между объектами.
    - Компонент Rigidbody определяет Color (цвет), Mode (режим), Intencity (интенсивность) освещения объектов в сцене. У каждого GameObject’а есть Transform.
    - Компонент Rigidbody определяет Position (положение), Rotation (вращение), и Scale (масштаб) каждого объекта в сцене. У каждого GameObject’а есть Transform.
32. Что такое альфа-канал (Alpha Channel)?
    - Альфа-канал — это дополнительный параметр террэйна (Terrain). Определяет высоту поверхности террэйна.
    - Альфа-канал — это дополнительный параметр материала (Material). Обычно определяет прозрачность цвета.
    - Альфа-канал — это четвертый компонент цвета в цифровых изображениях. Обычно определяет прозрачность цвета.
    - Альфа-канал — это канал связи, который используется Unity для связи с Asset Store.
33. Что такое скрипт (script)?
    - Текст, состоящий из команд, которые может обработать и выполнить компьютер
    - Последовательность выполнения команд, записанных в текстовых файлах.
    - Команды, которые выполняет компьютер
    - Запись в специальном текстовом редакторе
34. Как обнулить координаты, масштаб и поворот объекта на сцене?
    - В инспекторе компонента Transform установить параметры Position на 0
    - В инспекторе открыть контекстное меню компонента Transform нужного вам объекта
    - Перетащить объект в Scene View на нулевые координаты
    - Выделить объект, нажать Delete

35. Как импортировать 3D модель в Unity?
- Окно Hierarchy -> правая клавиша -> Import Model -> выбрать файл модели
- Окно Project -> правая клавиша -> Import Package -> выбрать файл модели
- Перетащить файл 3D модели в окно Project, или скопировать модель напрямую в папку Assets проекта
- Меню сверху -> GameObject -> Import Model -> выбрать файл модели

36. Когда выполняется метод Start() ?
- Каждый фиксированный промежуток времени
- Каждый раз при запуске сцены
- Каждый раз при запуске сцены, при каждом включении скрипта, при создании объекта с этим скриптом
- Каждый видимый кадр

37. Что делает команда Debug.Log()?
- Выводит сообщение в центр экрана
- Записывает сообщение в отдельном лог-файле
- Выводит сообщение в консоль
- Выводит сообщение в консоль И записывает сообщение в отдельном лог-файле

38. Когда выполняется метод Update() ?
- Каждый раз при выходе из программы
- Каждый раз при запуске программы
- Один раз при запуске сцены
- Каждый видимый кадр

39. Где узнать об ошибках компиляции?
- В консоли, нижняя часть интерфейса Unity
- Нажать кнопку Play, ошибка появится в центре экрана
- Посмотреть лог-файл в корне папки проекта
- Посмотреть консоль и лог-файл в IDE, который использовался для создания скрипта

40. Что такое переменная?
- Это второстепенный элемент скрипта, используемый для выполнения арифметических операций
- Это тип данных, меняющийся с течением времени
- Это любая часть скрипта, хранящая информацию

- Это подписанная область оперативной памяти компьютера, хранящая значение всевозможных типов данных

41. Что такое Vector3?
    - Это библиотека методов, необходимые для взаимодействия с физическим движком
    - Это форма конкатенации 3х переменных типа float
    - Это внутренняя структура данных Unity, созданная для комфортного манипулирования параметрами точек и векторов в 3х-мерном пространстве
    - Это базовый тип переменной в C#, созданный чтобы комфортно манипулирования параметрами точек и векторов в 3х-мерном пространстве
42. Что такое float?
    - Это базовый числовой тип переменной без плавающей запятой, содержит целое число
    - Это базовый числовой тип переменной с плавающей запятой, содержит число с десятичной дробной частью
    - Это специальный тип объекта, контейнер для компонентов Unity
    - Это специальный тип объекта, содержит координаты, поворот и масштаб игровых объектов Unity
43. Для чего нужна команда Destroy()?
    - Команда Destroy() удаляет GameObject
    - Команда Destroy() удаляет GameObject, компонент или ассет
    - Destroy() удаляет компонент
    - Destroy() удаляет ассет
44. Каково условие выполнение конструкции "if"?
    - Булево выражение в скобках должно равняться 0
    - Булево выражение в скобках должно равняться false
    - Булево выражение в скобках должно упрощаться до 0
    - Булево выражение в скобках должно равняться true
45. Выберите типы переменных в порядке: строковый, целочисленный, дробный.
    - string, int, float
    - string, float, int
    - float, integer, str
    - string, integer, float
46. Что из перечисленного НЕ настраивается компонентом Rigidbody?

- Сила гравитации
- Масса объекта
- Сила импульсов при столкновении
- Метод обработки столкновений

47. Какой метод нужно вызывать, чтобы передвигать объект вправо со скоростью speed?

- transform.position = new Vector3.MoveTowards(Vector2.right * speed * Time.deltaTime);
- gameObject.Transform.Vector3 = Vector2.right * speed * Time.deltaTime;
- transform.position = new Vector3(Vector2.right * speed * Time.deltaTime);
- transform.Translate(Vector2.right * speed * Time.deltaTime);

48. Массив letters содержит "A" в первом элементе, "B" - во втором, "C" - в третьем. Что выведет letters[1]?

- A
- B
- Ничего
- Будет ошибка

49. Какой класс отвечает за ввод (с клавиатуры или иным способом)?

- Input
- PlayerInput
- KeyCode
- Keyboard

50. Что даст объекту физический материл, в котором Friction установлен равным 0?

- Объект не сможет двигаться, если касается другого объекта
- Объект будет "скользкий"
- Объект будет прилипать к другим объектам
- Объект будет прыгучим

51. Что даст объекту физический материал, в котором Bounciness установлен равным 1?

- При касании вся сила другого объекта будет поглощаться
- При касании другому объекту будет отдаваться такая же сила, которая была приложена

- При касании другому объекту будет отдаваться половина той силы, которая была приложена
- При касании половина силы другого объекта будет поглощаться

52. Без чего пользовательский интерфейс (UI) не будет корректно работать?
- Main
- Button
- Panel
- Canvas

53. Какой метод кнопки использовать, чтобы задать действие при её нажатии?
- OnClick()
- OnButtonClick()
- OnGetClick()
- OnButtonDown()

54. Когда вызывается метод Update()?
- Каждую 1/60 секунды
- Перед началом FixedUpdate()
- Каждый кадр
- После завершения FixedUpdate()

55. Чему равно время между кадрами?
- 1/60 секунды
- Time.deltaTime
- Time.time
- Time.timeBetweenFrames

56. В каком разрешении компилируется игра для Android-устройств?
- .cs
- .bin
- .apk
- .exe