

ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
"Всероссийский детский центр "Океан"
(ФГБОУ "ВДЦ "Океан")

РАССМОТРЕНА
и рекомендована к реализации
методическим советом
отдела дополнительного
образования
управления образовательных
программ ФГБОУ "ВДЦ "Океан"
от 01.12 2023 г.
Протокол № 2

УТВЕРЖДЕНА
Директором
ФГБОУ "ВДЦ "Океан"
Соловей Н. В.
(приказ от 15. 01. 2024
№9-у)

ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА

«СТУДИЯ РАЗРАБОТКИ ИГР»

техническая направленность

Возраст обучающихся: 12 - 17 лет
Уровень освоения программы: базовый
Объем программы: 12 часов
Срок реализации: 21 календарный день

Автор-составитель:
Бондарюк Владимир Викторович,
педагог дополнительного образования,
ФГБОУ "ВДЦ "Океан"

г. Владивосток
2023 г.

ОГЛАВЛЕНИЕ

ЦЕЛЕВОЙ РАЗДЕЛ.....	6
1.1 Пояснительная записка.....	6
1.2 Цель и задачи программы.....	8
1.3 Планируемые результаты.....	9
СОДЕРЖАТЕЛЬНЫЙ РАЗДЕЛ.....	10
2.1 Содержание занятий программы.....	10
2.2 Сценарий разового занятия.....	11
ОРГАНИЗАЦИОННЫЙ РАЗДЕЛ.....	13
3.1 Учебный план.....	13
3.2. Календарный учебный график.....	14
3.3. Материально-технические условия реализации программы.....	16
3.4. Информационное обеспечение реализации программы.....	16
3.5. Кадровые условия реализации программы.....	16
3.6. Ресурсное обеспечение реализации программы.....	16
3.7. Учебно-методические материалы.....	17
3.8. Формы аттестации и демонстрации достижений.....	17
3.9. Оценочные материалы.....	18
3.10. Список литературы.....	18
ПРИЛОЖЕНИЕ.....	20
5.1. План-конспекты занятий.....	20
5.2. Глоссарий.....	27
5.3. Другие.....	Ошибка! Закладка не определена.
КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК.....	Ошибка! Закладка не определена.

ИНФОРМАЦИОННАЯ КАРТА ПРОГРАММЫ

Полное наименование программы	Дополнительная общеобразовательная общеразвивающая программа «Студия разработки игр»
Автор-составитель	Бондарюк Владимир Викторович., педагог дополнительного образования
Направленность	<i>Техническая направленность</i>
Вид деятельности	<i>Разработка игр/приложений/проектов с помощью мультиплатформенного движка Unity.</i>
Адресат программы	учащиеся 12 -17 лет
Срок реализации	1 смена (21 день)
Уровень программы	базовый
Объём программы	12 часов
Цель	Формирование у детей базовых представлений о Unity и программировании как о творческой деятельности через разработку приложений, компьютерных игр и мультимедийных проектов и повышение заинтересованности в изучении информационных технологий.
Задачи	<p>1. Обучающие:</p> <ul style="list-style-type: none"> – Познакомить с основными терминами связанных с игровой индустрией; – Познакомить с пулом профессий в игровой индустрии; – Познакомить алгоритмом работы кода; – Познакомить с понятиями программирования; – Познакомить с проектной деятельностью; – Изучить правила безопасности при работе с ЭВМ; – Познакомить с основными методами создания видео игры; – Знакомство со структурой и ведением проекта, умение его представлять. <p>2. Развивающие:</p> <ul style="list-style-type: none"> – Способствовать развитию внимания, наблюдательности и логики; – Содействовать расширению кругозора; – Развивать навыки печати на клавиатуре; – Развивать навыки работы с файловой системой ПК;

	<ul style="list-style-type: none"> – Развитие понимания необходимости обладать достаточными навыками владения информационными технологиями; – Развитие стремления полностью реализовывать свои проекты; – Развитие самостоятельности; – Развитие внимательности; – Развитие логического и творческого мышления. <p>3. Воспитывающие:</p> <ul style="list-style-type: none"> – Умение создавать коллективный продукт, помогать окружающим; – Развитие коммуникативных навыков; – Умение преодолевать трудности на пути достижения цели; – Способствовать развитию художественного вкуса; – Развитие внимательности.
Краткое содержание	<ol style="list-style-type: none"> 1) Введение в курс «Студия разработки игр». 2) Основы программирования на языке с# в Unity. 3) Создание прототипа 2D платформера. 4) Создание главного меню игры. 5) Создание новых механик. 6) Подведение итогов курса.
Планируемые результаты	<p>Личностные результаты:</p> <ul style="list-style-type: none"> – Развитие инженерного мышления, навыков программирования и гейм-дизайна; – Развитие навыков сотрудничества со сверстниками и взрослыми; – Развитие эстетического сознания через профессию левелдизайнера. <p>Метапредметные результаты:</p> <ul style="list-style-type: none"> – Владение обще предметными понятиями «проект», «идея», «алгоритм», «модель» и др.; – Владение информационно-логическими умениями: определять понятия, создавать обобщения, устанавливать аналогии, классифицировать, самостоятельно выбирать основания и критерии для классификации, устанавливать причинно-следственные связи, строить логическое рассуждение, умозаключение (индуктивное, дедуктивное и по аналогии) и делать выводы; <p>Предметные результаты:</p>

	<ul style="list-style-type: none"> – Владение основными терминами связанных с игровой индустрией; – Знания о профессиях в игровой индустрии; – Владение знаниями о алгоритме работы кода; – Владение знаниями о понятиях программирования; – Знания в проектной деятельности; – Знания правил безопасности при работе с ЭВМ; – Владение знаниями о основных методах создания видео игр; – Владение знаниями о структуре и ведения проекта, умение его представлять.
Социальный эффект	<p>В последнее время Unity развивается стремительными темпами, привлекая все больше новых разработчиков. На базе этого движка были созданы такие популярные игры, как Ori and the Blind Forest (разработчик: Moon Studios), Cities: Skylines (разработчик: Colossal Order), Dungeon of the Endless (разработчик: Amplitude Studios), а также мобильная версия Hearthstone: Heroes of Warcraft (разработчик: Blizzard Entertainment). Кроме того, Unity используется для создания приложений, связанных с архитектурой, обучением, визуализацией данных, электронными книгами и многими другими отраслями. Так при съемках фильма Книга джунглей (2016, режиссёр: Джон Фавро) приложение на базе Unity Engine использовалось для визуализации в режиме реального времени перемещения по сцене и работы с освещением.</p>
Год разработки	2020 - 2021
Год последней редакции	2023

1. ЦЕЛЕВОЙ РАЗДЕЛ

1.1 Пояснительная записка.

Данная программа разработана в соответствии с нормативными правовыми актами и государственными программными документами, являющиеся правовым основанием программы:

– Федеральный закон от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации»;

– Постановление Правительства Российской Федерации от 31.05.2019 № 694 «Об утверждении Устава Федерального государственного бюджетного образовательного учреждения «Всероссийский детский центр «Океан»»;

– Распоряжение Правительства Российской Федерации от 29.05.2015 № 996-р «Об утверждении Стратегии развития воспитания в Российской Федерации на период до 2025 года»;

– Распоряжение Правительства Российской Федерации от 31.03.2022 № 678-р «Об утверждении Концепции развития дополнительного образования детей и плана мероприятий по ее реализации»;

– Приказ Министерства просвещения Российской Федерации от 27.07.2022 N 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;

– Приказ Минпросвещения России от 3 сентября 2019 г. N 467 (ред. от 21.04.2023) «Об утверждении Целевой модели развития региональных систем дополнительного образования детей, утвержденная (Зарегистрировано в Минюсте России 06.12.2019 N 56722);

– Постановление Главного государственного санитарного врача Российской Федерации от 28.01.2021 N 2 «Об утверждении санитарных правил и норм СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания»;

– Постановление Главного государственного санитарного врача Российской Федерации от 28.09.2020 № 28 «Об утверждении санитарных правил СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи»;

– Программа развития федерального государственного бюджетного образовательного учреждения «Всероссийский детский центр «Океан» на 2021-2025 годы – утв. Врио директора ФГБОУ ВДЦ «Океан» 25.06.2021;

– Приказ ФГБОУ ВДЦ «Океан» от 06.09.2022 № 697-у «Об утверждении Положения о дополнительной общеобразовательной программе в ФГБОУ ВДЦ «Океан».

Дополнительная общеобразовательная общеразвивающая программа технической направленности «Студия разработки игр» (от английского game developer – разработчик игр).

Наше время диктует новые условия буквально в каждой области деятельности. Если компания хочет облегчить жизнь себе и своим клиентам, обойти конкурентов, то будет внедрять компьютерные технологии. Музыкальная индустрия и кино, мультипликация и игры, мобильная связь, банковское дело, дизайн и журналистика - все области тесно переплетены с программированием и используют специальные программы на разных этапах работы.

Знание основ программирования становится необходимым все большему количеству специалистов. В первую очередь, это, конечно, сами программисты, тестировщики программного обеспечения, системные администраторы, веб-мастера.

Во вторую - аналитики, менеджеры, маркетологи, которым знание азов написания скриптов может сэкономить много рабочего времени. Кроме того, знание программирования облегчает жизнь и людям таких творческих профессий, как дизайнеры, 3d-аниматоры и иллюстраторы. В сети можно найти много скриптов для графических программ, написанных именно этими творческими людьми.

В-третьих, знания основ программирования может пригодиться любому человеку. Когда возникает необходимость заказать сайт, внести дополнения в используемое для работы программное обеспечение, мы обращаемся к программистам. Чтобы на выходе работы программиста получить ровно то, что мы подразумевали, необходимо хотя бы владеть основными терминами информационных технологий и хорошо, если заказчик представляет, что действительно возможно сделать в конкретной ситуации и примерно за какой срок.

Робототехника тоже все больше входит в нашу жизнь. Роботов используют как на производстве, так и в медицине, и даже в обычной жизни. Чтобы робот делал то, что от него хотят, его сначала нужно запрограммировать. Программирование и создание игр с использованием инструмента разработки игр Unity, позволяет ребенку в увлекательной форме познакомиться с одним из популярных языков программирования C#. Наглядно показывает ребенку насколько удобны вдумчиво организованные каталоги папок и файлов, необходимых для разработки, нежели небрежно расположенные в одном месте файлы скриптов, изображений и файлы других типов.

На сайте Unity расположена довольно подробная документация, что дает возможность ознакомить детей с работой с документацией и приучать к самостоятельному освоению новых конструкций языков программирования. Обе среды дают возможность делиться своими проектами в международных сообществах, позволяют поддерживать интерес к информационным технологиям, желание узнавать больше и глубже вникать в предмет.

Unity являются бесплатными. Учащиеся могут заниматься техническим творчеством даже дома, что в свою очередь выводит их проекты на более высокий уровень.

Обучение всесторонне развивает ребенка: помогает правильно ставить цели, концентрироваться и работать на результат. А еще тренирует креативность, логику, умение общаться с другими ребятами и взрослыми, целеустремленность.

Ученики знакомятся, в команде решают задания и вместе обсуждают варианты. А сам процесс создания игры - очень увлекательный и интересный.

Программа имеет несколько модулей для «базового» и «продвинутого» уровня.

Уникальность программы заключается в том, что дети самостоятельно разрабатывают свою собственную игру.

Низкий порог вхождения, редактор движка имеет понятный, удобный и гибкий интерфейс. На официальном сайте Unity можно найти большое количество обучающего материала, как видео уроков, так и текстовых статей. Так же имеется официальная документация на русском языке, правда локализована только устаревшая версия (5.3), современные функции движка описаны только в англоязычной документации, но это не является большой проблемой. Движок поддерживает скриптовый язык: C#. Подавляющее большинство разработчиков используют популярный и современный язык программирования C#, работая при этом в удобном редакторе кода - Visual Studio, который, к слову, так же бесплатный. При желании можно пользоваться встроенным редактором кода MonoDevelop.

Программа рассчитана на шесть занятий по два академических часа. Есть возможность расширить за счет дополнительных модулей, представленных в Приложении в подразделе 5.3 Другое.

Целевой аудиторией данной программы являются ученики 7 – 11 классов (12 – 17 лет), проявляющие интерес к сфере разработки программного обеспечения при помощи языков программирования, изучению гейм дизайна.

Принцип набора – свобода выбора. Количество учащихся в группе: до 12 человек (ориентировано на количество рабочих мест, из расчета 1 компьютер = 1 ученик), состав может быть разновозрастной и разнополый.

Уровень программы:

«Базовый» объём – 12 часов. Срок реализации – 1 смена (21 день). Продолжительность занятия – 2 академических часа.

«Продвинутый» объём – 24 часа. Срок реализации – 1 смена (21 день)ю Продолжительность занятия – 2 академических часа.

1.2 Цель и задачи программы

Целью программы является формирование у детей базовых представлений о Unity и программировании как о творческой деятельности через разработку приложений, компьютерных игр и мультимедийных проектов и повышение заинтересованности в изучении информационных технологий.

- обучения работе с индивидуальными проектами от задумки до полной реализации в программном виде;
- обсуждения проектов и завершённых работ в группе.

Задачи:

1. Обучающие:

- Познакомить с основными терминами связанных с игровой индустрией;
- Познакомить с пулом профессий в игровой индустрии;
- Познакомить алгоритмом работы кода;
- Познакомить с понятиями программирования;
- Познакомить с проектной деятельностью;
- Изучить правила безопасности при работе с ЭВМ;
- Познакомить с основными методами создания видео игры;
- Знакомство со структурой и ведением проекта, умение его представлять.

2. Развивающие:

- Способствовать развитию внимания, наблюдательности и логики;
- Содействовать расширению кругозора;
- Развивать навыки печати на клавиатуре;
- Развивать навыки работы с файловой системой ПК;
- Развитие понимания необходимости обладать достаточными навыками владения информационными технологиями;
- Развитие стремления полностью реализовывать свои проекты;
- Развитие самостоятельности;
- Развитие внимательности;
- Развитие логического и творческого мышления.

3. Воспитывающие:

- Умение создавать коллективный продукт, помогать окружающим;
- Развитие коммуникативных навыков;
- Умение преодолевать трудности на пути достижения цели;
- Способствовать развитию художественного вкуса;
- Развитие внимательности.

1.3 Планируемые результаты

Личностные результаты:

- Развитие инженерного мышления, навыков программирования и гейм-дизайна;
- Развитие навыков сотрудничества со сверстниками и взрослыми;
- Развитие эстетического сознания через профессию левелдизайнера.

Метапредметные результаты:

- Владение обще предметными понятиями «проект», «идея», «алгоритм», «модель» и др.;
- Владение информационно-логическими умениями: определять понятия, создавать обобщения, устанавливать аналогии, классифицировать, самостоятельно выбирать основания и критерии для классификации,

устанавливать причинно-следственные связи, строить логическое рассуждение, умозаключение (индуктивное, дедуктивное и по аналогии) и делать выводы;

Предметные результаты:

- Владение основными терминами связанных с игровой индустрией;
- Знания о профессиях в игровой индустрии;
- Владение знаниями о алгоритме работы кода;
- Владение знаниями о понятиях программирования;
- Знания в проектной деятельности;
- Знания правил безопасности при работе с ЭВМ;
- Владение знаниями о основных методах создания видео игр;
- Владение знаниями о структуре и ведения проекта, умение его представлять.

2. СОДЕРЖАТЕЛЬНЫЙ РАЗДЕЛ

2.1 Содержание занятий программы

Занятие № 1

Тема: «Введение в курс «Студия разработки игр». Интерфейс. Возможности. Asset Store Unity»

Цель: Формирование мотивации учащихся к успешному освоению программы «Студия разработки игр». Знакомство с содержанием курса, а также основными инструментами для последующей работы.

Теория: Понятия «Программирование» и «Разработка игр». Знакомство с методологиями разработки, их принципами, особенностями и отличиями. Демонстрирования им успешных проектов разных известных разработчиков.

Практика: Знакомство с интерфейсом и функциональными характеристиками Asset Store и Unity.

Занятие № 2

Тема: «Основы программирования на языке C# в Unity.»

Цель: обеспечить углубление, обобщение и систематизацию знаний по программированию; способствовать формированию умения рационально подбирать методы разработки алгоритмов. Способствовать развитию аналитического и логического мышления, памяти, внимания, коммуникативных навыков участников программы;

Теория: Изучение принципов работы скрипта, функций и методов. Суть написания программ в Unity.

Практика: Написание и исправление готовых скриптов, написанных на языке программирования C#.

Занятие № 3

Тема: «Создание прототипа 2D платформера.»

Цель: научить принципам работы в Unity в 2D пространстве.

Теория: Объяснение принципов работы законов физики в Unity и задач скриптов. Работа с инструментом TileMap.

Практика: Применение и апробация полученных теоретических знаний. Создание уровня игры на основе сделанных физики и тайлмапа.

Занятие № 4

Тема: «Создание главного меню игры. Программирование кнопок в меню. Создание Scripts.»

Цель: создание главного меню игры, от которого будет стартовать проект.

Теория: Объяснение функциональных характеристик и назначение компонентов main_menu, изучение таких объектов как Button, Canvas и Text. Работа с кнопками и переключение сцен.

Практика: Создание скриптов на языке программирования C#. Создание странички автора. Применение и апробация полученных теоретических знаний.

Занятие № 5

Тема: «Разработка собственной 2D игры.»

Цель: Объединение наработанных ранее заданий в один целый проект.

Практика: Переключение между разными сценами. Перемещение персонажа. Создание наполнения игры.

Занятие № 6

Тема: «Завершение разработки собственной игр. Подведение итогов курса.»

Цель: Создание своей полноценной игры с Главным меню и страничкой автора.

Практика: Завершение работы над проектом, тестирование готовой игры, исправление недочетов, ошибок. Анкетирование проведение собеседования на добавление в telegram – канал, на должность разработчика в игровую студию ВДЦ «Океан», устная рефлексия.

2.2 Сценарий разового занятия

Тема занятия:

«Создание графики для компьютерной игры»

Цели занятия:

Познакомить детей с базовыми представлениями о профессии 3D-артист по средствам программы MagicaVoxel.

Задачи:

- обучающая:

Познакомить детей с понятие Voxel, где можно применять объекты созданные в программе;

- развивающая:

Развитие навыков работы с наглядным примерами, самостоятельная работа над поставленной задачей, работы с прикладной программой;

Тип занятия: комбинированный занятие, совмещает практическую часть и теоретическую

Форма организации работы на занятии:

- индивидуальная работа;

Основные методы, применяемые на занятии:

- словесные методы;
- наглядные методы;
- практические методы.

Оборудование занятия:

а) Компьютеры для каждого участника занятия

б) Установленное ПО (MagicaVoxel)

Ход занятия, его содержание:

- Организационный момент - 2-3 минуты.

Цель: подготовка учащихся к работе на занятии.

Педагог приветствует учеников, получает список присутствующих на занятии, проводит инструктаж пожарной безопасности и на рабочем месте.

- Подготовка к изучению нового материала.

Цель: формирование познавательных мотивов, подготовка учащихся к восприятию нового материала, раскрытие темы и цели занятия.

Педагог: Наши занятия посвящены разработки игр, но для того чтобы ваша игра выглядела красиво, в ней должна присутствовать приятная глазу графика. MagicaVoxel позволяет создать модели / объекты в оригинальном стиле Voxel. Все слышали, о таком стиле искусства как PixelArt, Voxel это тот же “пиксель”, но уже в трехмерном пространстве. Вы сможете увидеть свою созданную модель со всех сторон. Также программа позволяет конвертировать созданную модель в популярные форматы для различных программ для дальнейшей работы. Формат fbx позволяет нам переносить объекты в различные программы для 3d моделинга, таковыми является unity, blender и многие другие.

Сегодня вы создадите сцену из популярной игры “Crossy road” Игра была популярна долгое время из-за своей оригинальной графики, в оригинальном стиле.

- Изучение нового материала

Педагог: запускаем программу MagicaVoxel, перед вами стартовая сцена любого проекта. Навигация по сцене осуществляется посредством нажатия правой кнопки мыши мы вращаем объект при помощи колесика мыши можем уменьшать или увеличивать масштаб.

Слева присутствует окно палитра здесь мы задаем нужный нам цвет

Далее познакомимся с кистями: в программе есть 4 кисти 2 вспомогательные и 2 основные. К вспомогательным относятся линия и точка к основным плоскость и заданная область по плоскости. Любая из кистей имеет 3 функции Attach Erase и Paint.

- Attach - служит для наращивания вокселей
- Erase - служит для вдавливания / удаления верхнего слоя вокселей
- Paint - служит для перекрашивания выбранного верхнего слоя вокселей.

Также во вкладке Render вы сможете увидеть свою модель с освещением также вы сможете придать свойства различного материала вашим объектам
Заключение:

Как вы поняли программа очень проста и интуитивна понятна. Теперь ваша задача создать свой арт в стиле Voxel, как пример могу вам дать вот эту картинку

3. ОРГАНИЗАЦИОННЫЙ РАЗДЕЛ

3.1 Учебный план

Таблица № 1 Учебный план программы для «Базового» уровня

№	Тема	Кол-во часов			Форма аттестации/ контроля
		всего	теория	практика	
1	Введение в курс «Студия разработки игр» Интерфейс. Возможности Asset Store Unity.	2	1,5	0,5	Опрос. Вводное анкетирование. (В Google форма) Включенное наблюдение
2	Основы программирования на языке с# в Unity.	2	1,5	0,5	Результаты заданий, самодиагностика, диагностика валидными методами
3	Создание прототипа 2D платформера	2	0,5	1,5	
4	Создание главного меню игры. Программирование кнопок в меню. Создание Scripts	2	0,5	1,5	
5	Разработка собственной 2D игры.	2	-	2	
6	Завершение разработки собственной игр. Подведение итогов курса	2	-	2	Собеседование (Экзамен). Разработанные и защищенные проекты, рефлексия.

3.2. Календарный учебный график

№	Сроки проведения смены	Программы смены, в рамках которых может реализовываться ДООП «Студия разработки игр»	Кол-во часов в неделю	Кол-во учебных дней в смене
1	9 - 29 января	«Творчество без границ» (инженерно-технический модуль), «#ПроСпорт», Модуль «Мир шахмат»	6 ак. часов в неделю	12 учебных д ней
2	1 - 21 февраля	Фестиваль «Океанские подмостки», Модуль «Отличники первых», Фестиваль науки «На пороге открытий» Модуль «Изобретатели будущего», Модуль «Микромир на ладони»	6 ак. часов в неделю	12 учебных д ней
3	26 февраля - 17 марта	«Зимняя спортивно-образовательная школа „Мастерская физоргов“», «Творческий слёт „В ритме „Океана“», «Океанские принты», Модуль «Отличники первых», Модуль «Курс на взлёт»	6 ак. часов в неделю	12 учебных д ней
4	12 апреля - 2 мая	«Счастливым май», Модуль «Без срока давности», Модуль «Историко-патриотическая смена „Школьный музей Победы“», Модуль «Отличники первых», Модуль «Потомки победителей», «На волне спорта», «Искусство кода»	6 ак. часов в неделю	12 учебных д ней
5	5 - 18 мая	Содружество орлят России	6 ак. часов в неделю	12 учебных д ней
6	27 мая - 16 июня	«Здоровое поколение», Модуль «ГородОКеан», «Волонтеры финансового просвещения», «Школа игропрактиков», «Россия 7мастеровая», «#ВеликийМогучий», Модуль «Отличники первых», «Тихоокеанская школа безопасности», «Международный детский медиасаммит», «Туризм и гостеприимство», «Моя первая книга», «По морям вокруг Земли»	6 ак. часов в неделю	12 учебных д ней
7	20 июня - 10 июля	«Экологический форум „Живи, Земля!“», Модуль «Таинственный микромир», «Дальневосточный рубеж 2.0», «Океанский марафон активностей», «Техноканикулы», Модуль «Отличники первых», «Туризм и гостеприимство», «Мастера событий», «Летняя арт-деревня», «Мои возможности», «Дороги без опасности», «Морской старт»	6 ак. часов в неделю	12 учебных д ней
8	14 июля - 3 августа	«Дети Мира» (художественный модуль), «Дети Мира» (физкультурно-спортивный модуль),	6 ак. часов в неделю	12 учебных д ней

		Модуль «Отличник первых», «Слёт школьных лесничеств „В защиту леса“», «Исследователи моря», «Сила моря»		
9	7 - 27 августа	«Планета цирка», «Страна железных дорог», «Детская лаборатория туризма „Родные маршруты“», «Океанский марафон активностей», «Проба в профессиях», Модуль «Финансовая академия FINOcean», Модуль «Курс на бизнес и предпринимательство», Модуль «Отличники первых», Модуль «Юный следователь», Модуль «Юный казначей», Модуль «Юный правовед», Модуль «Мы – государство», «Бизнес-лагерь», «Океан талантов», «Летняя арт-деревня», «Наследники Петра I», Модуль «Будущее в науке»	6 ак. часов в неделю	12 учебных д ней
10	30 августа - 19 сентября	«Полуфинал Всероссийского конкурса „Большая перемена“», «IV Всероссийский фестиваль-конкурс детских духовых оркестров „Дальневосточные фанфары“», «Мир открытий», «Восточный ветер», Модуль «X Всероссийский сбор юных моряков „Юнга“», Слёт кадетских корпусов и классов «Служить России!», Модуль «Отличники первых», «На страже Родины», «Дороги памяти»	6 ак. часов в неделю	12 учебных д ней
11	27 сентября - 17 октября	«Мой край, горжусь тобой!», «Экологика», «ЭтноРоссия», «#БлизкийДальний»	6 ак. часов в неделю	12 учебных д ней
12	20 октября - 9 ноября	«Пушкинская осень в „Океане“», Модуль «Знание. Авторы», «Инновациям – старт!» (техника), Модуль «Как это работает: машиностроение», Модуль «Море возможностей», Модуль «Умный город 4.0 (автономный и беспилотный транспорт)», Модуль «В мире судостроения», Модуль «Энергия старта», Модуль «Отличники первых»	6 ак. часов в неделю	12 учебных д ней
13	12 ноября - 2 декабря	«Вектор успеха», «Инновациям – старт!» (наука), Модуль «Про Е и не только (пищевые и биологически активные добавки)», Модуль «Вырасти свою бактерию», Модуль «Школа инженерных решений», Модуль «Получение и свойства наночастиц и наноматериалов»	6 ак. часов в неделю	12 учебных д ней
14	5 - 25 декабря	«Дом культуры», «Зимняя спортивно-образовательная школа	6 ак. часов в неделю	12 учебных д ней

		„Мастерская физоргов“», Модуль «Отличники первых»		
--	--	--	--	--

3.3. Материально-технические условия реализации программы

Для успешной реализации программы необходимо следующее *материально-техническое обеспечение*:

- помещение для постоянных занятий с хорошим освещением и доступом к сети Интернет;
- столы и стулья согласно ГОСТу и списочному составу учащихся;
- рабочие места учащихся: компьютеры с установленным ПО, соответствующим тематике курса (Unity);
- рабочее место педагога: компьютер, интерактивная доска (экран), проектор.

Оснащение компьютерной техникой должно производиться из расчета: 1 участник программы = 1 рабочее место, т.к. работа по два или более учащихся за одним компьютером существенно снизит не только удовлетворенность участников, но и результативность реализации программы.

3.4. Информационное обеспечение реализации программы

Для реализации программы рекомендованы следующие информационные ресурсы:

Сайт документации Unity <https://docs.unity3d.com/Manual/> так как весь материал изложен на английском языке. Так же рекомендуется следующий российский ресурс <https://docs.unity3d.com/ru/530/Manual/UnityManual.html> , но, к сожалению, информация на этом ресурсе давно не обновлялась.

Сайт для изучения языка программирования C# <https://metanit.com/sharp/> сайт подходит, как и новичкам так и опытным программистам.

3.5. Кадровые условия реализации программы

Педагог со средним специальным или высшим педагогическим образованием. Знание языка программирования C# на уровне Junior+ или Middle. Знание игрового движка Unity уровне Junior+ или Middle.

3.6. Ресурсное обеспечение реализации программы

Для обеспечения правильной технологии выполнения программы понадобятся следующие ресурсы:

- 1) Педагог дополнительного образования со средним или высшим

образованием.

- 2) Кабинет площадью 30 м² или более.
- 3) Интернет с пропускной способностью не меньше 100 Мб/сек на рабочее место.

3.7. Учебно-методические материалы

UNITY и C# ГЕЙМДЕВ от ИДЕИ до РЕАЛИЗАЦИИ - Unity и C#. Геймдев от идеи до реализации. 2-е изд.. — СПб.: Питер, 2019. ISBN 978-5-4461-0715-5.

ГЕЙМДИЗАЙН, как создать игру в которую будут играть все - Джесси Шелл

Unity в действии – Джозеф Хокинг - Unity в действии. Мультиплатформенная разработка на C#. 2-е межд. изд. — СПб.: Питер, 2019. — 352 с.: ил. — (Серия «Для профессионалов»). ISBN 978-5-4461-0816-9.

Развитие эмоционального интеллекта у детей и взрослых - ISBN 978-5-907308-08-4

Проект за 10 шагов: навигатор для организации проектной деятельности подростков / Косолапова Ю. В., Морозова С. Ю. – Воронеж: ООО «Алекс Принт», 2021. – 96 с. ISBN 978-5-6047021-3-0.

Одной из приоритетных задач программы является формирование и развитие у детей навыков работы с поиском, анализом и структурированием информации из различных интернет-источников, в связи с этим изложение учебного материала имеет некоторые особенности. Педагог в беседе дает лишь небольшую часть информации, основную информацию учащиеся добывают в процессе глобальной сети.

Формы организации учебной деятельности: индивидуальная; групповая; фронтальная; выставка.

Основной вид занятий – практический.

В качестве методического обеспечения программы педагог использует:

- беседы: «Расскажи о себе», «Что такое алгоритм? Где в жизни мы встречаем примеры алгоритмов?», «Что такое программирование и чем оно отличается от разработки приложений»;
- демонстрационные приложения участников программы других смен.

3.8. Формы аттестации и демонстрации достижений

Формы аттестации и демонстрации достижений: творческая работа, защита проекта, участие в выставке детского творчества.

Формы фиксации образовательных результатов: журнал посещаемости, фото, отзывы детей.

Оформление результатов исследования: аналитическая записка по материалам диагностических мероприятий.

3.9. Оценочные материалы

При успешном освоении программы, и участии в Итоговой выставке с презентацией выполненной итоговой работой, учащиеся получают установленную на свой девайс игру для дальнейшего использования, а также сертификат об окончании программы.

В случае пропуска более 2 занятий, программа считается не освоенной, сертификат не полагается. Но учащиеся забирают свою игру.

Одним из показателей результативности является сохранность созданного контента, а также повышение самооценки учащихся, их личностные приращения, удовлетворенность программой, эмоциональное состояние участников программы.

Эти показатели определяются с помощью пакета диагностических методик, в который входят: анкетирование, тестирование, методика незаконченных предложений, методика изучения мотивации обучения у подростков.

Критерии и показатели уровня усвоения программы:

Низкий уровень – на базе Unity учащиеся умеют создавать проект, используя подсказки или с помощью педагога. Не умеет искать и структурировать информацию.

Средний уровень – на базе Unity учащиеся умеют создавать проект с небольшим наполнением, но выполняют это самостоятельно, опираясь на полученные ранее знания, не используя дополнительные подсказки и не обращаясь за дополнительной консультацией к педагогу.

Высокий уровень – на базе Unity учащиеся умеют создавать более сложные Script'ы, использовать различные компоненты, изученные ранее. Самостоятельно прорабатывают алгоритм работы программы и рисуют сцену с наполнением, могут рассказать и объяснить другим участникам или посторонним людям, о чем его продукт и как он функционирует, а также какие компоненты и почему он использует.

Механизм отслеживания результатов реализации программы:

- 1) начальный контроль (вводное анкетирование, беседа);
- 2) текущий контроль (осуществляться по результатам выполнения учащимися практических заданий);
- 3) промежуточный контроль (выполнение творческих заданий, самостоятельных работ, промежуточное анкетирование);
- 4) итоговый контроль (защита проектов, выставка работ, итоговое анкетирование).

3.10. Список литературы

1. Джесси Шелл Гейм дизайн – Москва.: альпина паблишер, 2020 – 639 с.
2. Джозеф Хокинг Unity в действии – СПб.: Питер, 2020 – 352 с.

3. <http://window.edu> (Единое окно доступа к образовательным ресурсам)
4. <http://www.edu.ru> (Федеральный портал «Российское образование»)
5. <http://school.edu.ru> (Российский общеобразовательный портал)
6. <http://ege.edu.ru> (Портал информационной поддержки единого государственного экзамена)

Список рекомендованной литературы для преподавателя:

Основная:

Линовес Д. Виртуальная реальность в Unity. - М.: ДМК Пресс, 2015. - 194 с.

Джесси Шелл Гейм дизайн – Москва.: альпина паблишер, 2020 – 639 с.

Джозеф Хокинг Unity в действии – СПб.: Питер, 2020 – 352 с.

Дополнительная:

Гин А. А. Приёмы педагогической техники: свобода выбора, открытость, деятельность, обратная связь, идеальность: Пособие для учителей. — Гомель: ИПП «Сож», 1999. — 88 с.

Бреннан К. Креативное программирование. - М. Чунг. — Гарвардская Высшая школа образования, 2017.

Список рекомендованной литературы для обучающихся:

Основная:

Курс линейной алгебры (<https://stepik.org/course/2461>).

Основы Git (<https://stepik.org/course/3145>).

Программирование на C# (<https://stepik.org/course/4143>).

Официальный курс Unity (<https://unity.com/ru/learn>)

Дополнительная:

<http://www.virryvr.com/>

<https://www.labster.com>

<https://itunes.apple.com/us/app/jurassic-virtual-reality-vr/id958174054?mt=8>

<https://edu.google.com/expeditions>

<http://www.titansofspacevr.com/>

<http://anatomy4d.daqri.com/>

<http://elements4d.daqri.com/>

<https://www.microsoft.com/en-us/store/p/mylab/9nn8dz3j8ksx>

<https://itunes.apple.com/us/app/ar-planets/id839735420?mt=8>

<https://play.google.com/store/apps/details?id=com.google.android.stardroid&hl=en>

4. ПРИЛОЖЕНИЕ

4.1. План-конспекты занятий

Тема занятия:

“Создание графики для компьютерной игры”

Цели занятия:

Познакомить детей с базовыми представлениями о профессии 3D-артист по средствам программы MagicaVoxel.

Задачи:

- *обучающая:*

Познакомить детей с понятие Voxel, где можно применять объекты созданные в программе;

- *развивающая:*

Развитие навыков работы с наглядным примерами, самостоятельная работа над поставленной задачей, работы с прикладной программой;

Тип занятия: комбинированный занятие, совмещает практическую часть и теоретическую

Форма организации работы на занятии:

- индивидуальная работа;

Основные методы, применяемые на занятии:

- словесные методы;
- наглядные методы;
- практические методы.

Оборудование занятия:

а) Компьютеры для каждого участника занятия

б) Установленное ПО (MagicaVoxel)

Ход занятия, его содержание:

- Организационный момент - 2-3 минуты.

Цель: подготовка учащихся к работе на занятии.

Педагог приветствует учеников, получает список присутствующих на занятии, проводит инструктаж пожарной безопасности и на рабочем месте.

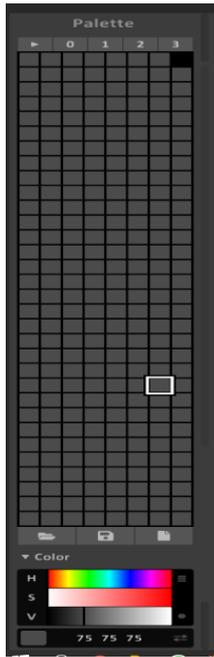
- Подготовка к изучению нового материала.

Цель: формирование познавательных мотивов, подготовка учащихся к восприятию нового материала, раскрытие темы и цели занятия.

Педагог: Наши занятия посвящены разработки игр, но для того чтобы ваша игра выглядела красиво, в ней должна присутствовать приятная глазу графика. MagicaVoxel позволяет создать модели / объекты в оригинальном стиле Voxel. Все слышали, о таком стиле искусства как PixelArt, Voxel это тот же “пиксель”, но уже в трехмерном пространстве. Вы сможете увидеть свою созданную модель со всех сторон. Также программа позволяет конвертировать созданную модель в популярные форматы для различных программ для дальнейшей работы. Формат fbx позволяет нам переносить объекты в

различные программы для 3d моделинга, таковыми является unity, blender и многие другие.

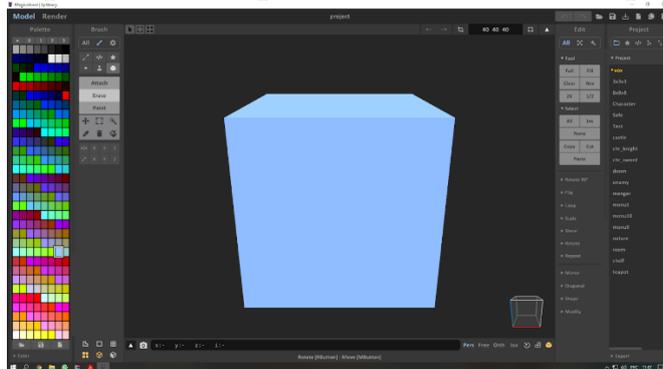
Сегодня вы создадите сцену из популярной игры “Crossy road”



Игра была популярна долгое время из-за своей оригинальной графики, в оригинальном стиле.

- Изучение нового материала

Педагог: Запускаем программу MagicaVoxel, перед вами стартовая сцена любого проекта,



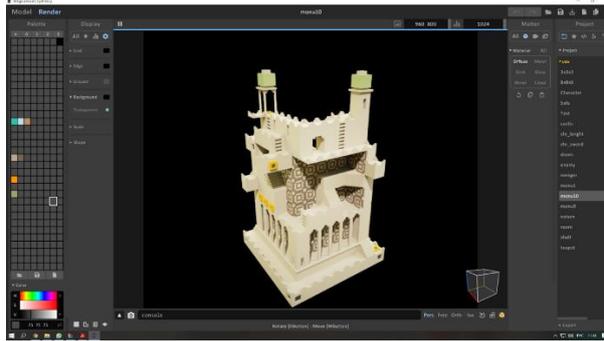
Навигация по сцене осуществляется посредством нажатия правой кнопки мыши мы вращаем объект при помощи колесика мыши можем уменьшать или увеличивать масштаб.

Слева присутствует окно палитра здесь мы задаем нужный нам цвет Далее познакомимся с кистями: в программе есть 4 кисти 2 вспомогательные и 2 основные. К вспомогательным относятся линия и точка к основным плоскость и заданная область по плоскости. Любая из кистей имеет 3 функции Attach Erase и Paint



- *Attach - служит для наращивания вокселей*

- *Erase* - служит для вдавливания / удаления верхнего слоя вокселей
- *Paint* - служит для перекрашивания выбранного верхнего слоя вокселей



Также во вкладке *Render* вы сможете увидеть свою модель с освещением также вы сможете придать свойства различного материала вашим объектам

Заключение:

Как вы поняли программа очень проста и интуитивна понятна. Теперь ваша задача создать свой арт в стиле Voxel, как пример могу вам дать вот эту картинку



Тема занятия: **"Базовый двумерный платформер."**

Цели занятия:

Познакомить детей с базовыми представлениями о геймдизайне и программированию.

Задачи:

- *обучающая:*

Познакомиться с понятием графического режима в двумерном пространстве, анимация на базе листов спрайтов, физические явления в двумерном пространстве (столкновения, гравитация), управление камерой в сайд-скроллерах;

- *развивающая:*

Развитие навыков работы с наглядными примерами, самостоятельная работа над поставленной задачей, работы с прикладной программой;

Тип занятия: комбинированное занятие, совмещает практическую часть и теоретическую

Форма организации работы на занятии:

- индивидуальная работа;
- групповая работа.

Основные методы, применяемые на занятии:

- словесные методы;
- наглядные методы;
- практические методы.

Оборудование занятия:

а) Компьютеры для каждого участника занятия

б) Установленное ПО (Unity, vs Code)

Ход занятия, его содержание:

- **Организационный момент - 2-3 минуты.**

Цель: подготовка учащихся к работе на занятии.

Педагог приветствует учеников, получает список присутствующих на занятии, проводит инструктаж пожарной безопасности и на рабочем месте.

- **Подготовка к изучению нового материала.**

Цель: формирование познавательных мотивов, подготовка учащихся к восприятию нового материала, раскрытие темы и цели занятия.

Педагог: Наши занятия посвящены разработке игр с помощью Unity. Unity – это межплатформенная среда разработки компьютерных игр. Проще говоря игровой движок. Давайте рассмотрим ТОП-7 лучших игр на Unity, чтобы убедиться в силе и способностях данного движка.

Hearthstone: Heroes of Warcraft - карточная онлайн-игра по мотивам вселенной *Warcraft*, *Cities Skylines* – симулятор градостроителя, *Cuphead* – довольно сложный платформер, *Pokemon Go*, *Subnautica*. И многое другое и если вы продолжите изучать гейм дизайн вы возможно станете одним из разработчиков какой либо известной игры, такой как *Cyberpunk 2077* =)

Сегодня мы научимся, создавать 2D платформер, создадим сцену с персонажем и платформами, узнаем, что такое спрайты, познакомимся с таким языком программирования как C#. Напишем код для перемещения персонажа по сцене. Запустим анимацию персонажа. И скомпилируем готовый проект.

- **Изучение нового материала**

Цель: Создать свой проект под именем «...», дать учащимся общие сведения об интерфейсе Unity, загрузить спрайты, создать уровень игры, добавить на него персонажа.

Педагог: Первым делом нам нужен двумерный проект. Выбираем команду New которая появляется после запуска Unity. Или команду New Project в меню File. Выбираем вариант 2D создает внутри проекта две папки с именем Sprites и Scripts. В них мы будем складывать в них различные

ресурсы. Для это во вкладке проекта нажмите правой кнопкой мыши, create, folder. Далее импортируем несколько спрайтов(изображений) (В двумерных играх изображения называются спрайтами, а в трехмерных текстурами) и перетащите эти изображения на сцену. Расположите Block в точке с координатами 0,15, -1,27, 0, а в полях масштаба введите значение 2, 1, 1. Присвойте спрайту имя Floore. Перетащите в сцену еще один спрайт Vox, поместите его на платформу и задайте размер 0,5; 0,5; 1

Педагог: Ваша задача добавить персонажа stickman самостоятельно, расположить его на сцене и переименовать объект в Player.

Ученики: Добавляют спрайты персонажа из указанной папки.

4. Смещение персонажа вправо и влево

Педагог: Для того, чтобы наш персонаж начал «видеть» препятствия, обрывы, начал существовать на сцене как игровой объект, а не спрайт к нему нужно добавить пару компонентов. А именно Rigidbody (риджидбоди) и Collider (коллайдер). Первый указывает симулятору физики, на какие объекты он должен действовать, а второй форму на которую будут действовать законы физики. Какой закон физики не дает нам сейчас улететь в небо

Ученики: Закон всемирного тяготения.

Педагог: Теперь чтобы наш персонаж ни куда не улетел мы отключаем силу тяжести для него. В параметрах Rigidbody находим строчку Gravity Scale и ставим значение 0. Также для параметра Collision Detection значение Continuous. И запрещаем вращение по оси Z.

Педагог: Заходим в ранее созданную папку скриптов создаем в ней скрипт C# (creat => C# Scripts). Запускаем файл. В нем мы видим уже есть написанный код. Это два метода Start и Update. Различаются они тем что во время запуска игры метод start запустится всего один раз при запуске игры, а метод апдейт будет запускаться каждый кадр.

Пример кода:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class Platformer_Player : MonoBehaviour{
    private float deltaX;// Создаем необходимые переменные для хранения
данных в памяти
    public float speed = 250.0f;
    private Rigidbody2D _body;

    void Start() {
        _body = GetComponent<Rigidbody2D>(); // в переменную хранящую
компонент риджитбади помещаем риджитбади игрока }
    void FixedUpdate(){
        deltaX = Input.GetAxis("Horizontal") * speed;
        Vector2 MOVEMENT = New Vector2(deltaX, _body.velocity.y);
```

```

    _body.velocity = movement;// Задаем только горизонтальное
    движение, сохраняем заданное вертикальное смещение
  }}

```

Педагог: Запускаем игру и видим, как персонаж по нажатию клавиш AD или стрелок начинает двигаться влево и вправо. Корректируем скорость.

5. Система анимации Mecanim

Цель: Создать анимированные движения при перемещении персонажа по сцене.

Педагог: Mecanim – это название системы анимации в юнити. Она позволяет полностью настроить систему анимированных клипов и обойтись минимум кодом при управлении этими клипами. В основе системы анимации лежат два вида ресурсов: анимированные клипы и анимированные контроллеры. Клипы представляют собой отдельные циклы анимации, в то время как контроллер – это сеть, определяющая, когда будет воспроизводиться каждый клип. Для того что бы создать анимацию выделите кадры 0-1 перетащите их на сцену и укажите имя в открывшемся окне имя stickman_idle. На вкладке с ресурсами появился клип stickman_idle и контроллер stickman_0 переименуйте контроллер в stickman. Мы только что создали анимацию для стоящего персонажа (удалить появившейся в сцене объект stickman_0). Повторите тот же процесс для создания анимации ходьбы и назовите ее stickman_walk. Далее добавляем компонент Animator на персонажа и в графу Controller перетаскиваем, созданный нами контроллер (по умолчанию клип со статическим состоянием воспроизводится очень быстро, поэтому выделите блок для этого состояния и на панели справа введите в поле Speed значение 0,2.

Педагог:

Each of these blocks is an animation "state".

Перейдите на вкладку Parameters

Нажмите + добавьте параметр типа Float с именем speed

После запуска компонента аниматор переключается между состояниями

Снимите флажок

Щелкните на состоянии при кнопкой мыши и выберите команду Make Transition. Повторить и в обратном направлении.

перехода. В нашем случае переход будет возникать при скорости больше 0,1

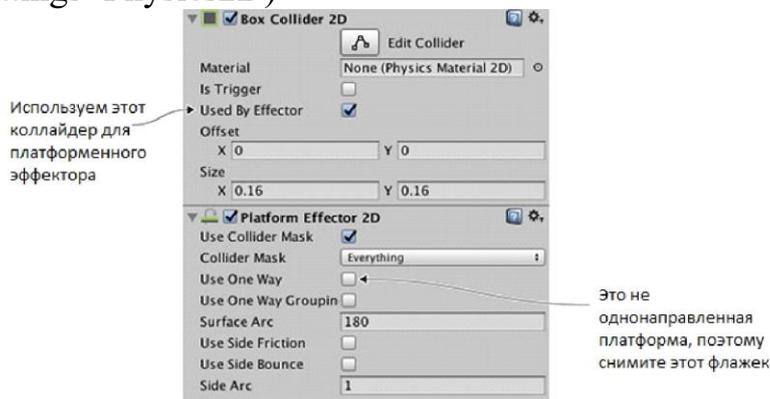
Педагог: Далее открываем скрипт персонажа и добавляем пару строчек в скрипт

```
...
private Animator _anim;
...
void Start() {
    _body = GetComponent<Rigidbody2D>();
    _anim = GetComponent<Animator>();
}
void Update() {
    ...
    _anim.SetFloat("speed", Mathf.Abs(deltaX));
    if (!Mathf.Approximately(deltaX, 0)) {
        transform.localScale = new Vector3(Mathf.Sign(deltaX), 1, 1);
    }
}
```

Педагог: Теперь мы видим как персонаж шевелиться при ходьбе.

6. Падение под силой тяжести и прыжок

Педагог: На первый это кажется не логичным, но прежде чем научить персонажа прыгать, следует сделать так что бы он падал. Меняем параметр Gravity Scale на 1 в компоненте Rigidbody2D. (Если нужно меняем настройки физики Project Settings>Physics2D)



Педагог: Теперь немного покодим:

```
...
public float jumpForce = 12.0f;
...
_body.velocity = movement;
if (Input.GetKeyDown(KeyCode.Space)) {
    _body.AddForce(Vector2.up * jumpForce, ForceMode2D.Impulse);
}
```

Педагог: И еще чуток покодим:

```
...
private BoxCollider2D _box;
...
_box = GetComponent<BoxCollider2D>();
...

```

```

_body.velocity = movement;
Vector3 max = _box.bounds.max;
Vector3 min = _box.bounds.min;
Vector2 corner1 = new Vector2(max.x, min.y - .1f);
Vector2 corner2 = new Vector2(min.x, min.y - .2f);
Collider2D hit = Physics2D.OverlapArea(corner1, corner2);
bool grounded = false;
if (hit != null) {
    grounded = true;}
...
if (grounded && Input.GetKeyDown(KeyCode.Space)) {

```

7. Заключение

Педагог: Теперь мы соберем наш проект в одну папку и сможем его запускать на любом компьютере, конечно если на нем установлен Windows). File>Build Settings>Build.

И так наше занятие подошло к концу. Я очень рад что у вас все получилось, хотелось бы выслушать ваши предложения по тому что можно добавить в этот урок или что убрать. Что вы думаете по поводу гейм дизайна в целом. Будите ли вы дальше узнавать эту программу работать в ней и развиваться как гейм дизайнер?

5.2. Глоссарий

Студия разработки игр. – от англ. сокращение от *Game Development* – разработка игры, на игровом движке Unity

Unity – межплатформенная среда разработки игр.

Алгоритмизация – процесс составления алгоритмов для решения поставленных прикладных задач.

Алгоритм – это точный набор инструкций, описывающих порядок действий некоторого исполнителя для достижения результата, решения некоторой задачи за конечное число шагов.

Asset Store – это растущая библиотека ассетов.

Ассет или игровой ресурс - цифровой объект, преимущественно состоящий из однотипных данных, неделимая сущность, которая представляет часть игрового контента и обладает некими свойствами.

Prefab – Это особый тип ассетов, позволяющий хранить весь GameObject со всеми компонентами и значениями свойств.

GameObject – Любой объект находящийся на сцене игры.

Открытое программное обеспечение – распространяющееся бесплатно.

Рефлексия – это обращение внимания субъекта на самого себя и на своё сознание, в частности, на продукты собственной активности, а также какое-либо их переосмысление.

Система команд исполнителя – это все команды, которые исполнитель умеет выполнять.

Среда исполнителя – обстановка, в которой функционирует исполнитель.

Технологический уклад – совокупность сопряжённых производств, имеющих единый технический уровень и развивающихся синхронно/

Фронтальная форма организации учебной деятельности – такой вид деятельности учителя и учащихся на уроке, когда все ученики одновременно выполняют одинаковую, общую для всех работу, всем классом обсуждают, сравнивают и обобщают результаты ее.