


Правительство Российской Федерации
Министерство просвещения Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
"Всероссийский детский центр "Океан"

Принята на заседании методического совета

Протокол № 1 от 12.01.2022 г.

СОГЛАСОВАНО

Начальник управления общего и
дополнительного образования
ФГБОУ "ВДЦ "Океан"

 М. И. Фролова
«19» января 2022 г.

УТВЕРЖДАЮ

Заместитель директора
по образовательной деятельности
ФГБОУ "ВДЦ "Океан"

 Г. Г. Рыбкин
«24» января 2022 г.

**Дополнительная
общеобразовательная общеразвивающая программа
технической направленности
«Game_Dev»**

Возраст учащихся – 12–17 лет
Срок реализации – 1 смена (21 день)
Количество часов – 12

Автор-составитель:
Бондарюк Владимир Викторович,
педагог дополнительного образования

Владивосток, 2022 г.

Владивосток, 2021 г.

ИНФОРМАЦИОННАЯ КАРТА ПРОГРАММЫ

Полное наименование программы	Дополнительная общеобразовательная общеразвивающая программа «Game_dev»
Автор-составитель	Бондарюк Владимир Викторович., педагог дополнительного образования
Направленность	<i>Техническая направленность</i>
Вид деятельности	<i>Разработка игры на мультиплатформенном движке Unity.</i>
Адресат программы	учащиеся 12 -17 лет
Срок реализации	1 смена (21 день)
Уровень программы	стартовый
Объём программы	12 часов
Цель	Целью программы является развитие инженерного мышления, творческих и проектных компетенций в процессе разработки игр на базе Unity в условиях лаборатории технического творчества.
Задачи	<ul style="list-style-type: none">-создать условия для освоения обучающимися базовых технических и специальных инженерных компетенций в процессе разработки игр.-содействовать развитию творческих и проектных компетенций в разноформатных активностях технического и цифрового творчества.-организовать пространство коммуникации, способствующее расширению опыта взаимодействия для решения инженерно-технических задач и саморазвития.
Краткое содержание	1)

<p>Планируемые результаты</p>	<p>Образовательными продуктами (внешними) станут технические идеи и прототипы игр. Образовательными результатами (внутренними) будут:</p> <p><i>Предметные:</i> Базовые технические компетенции в знании и умении.</p> <p><i>Метапредметные:</i></p> <ol style="list-style-type: none"> 1. Регулятивные универсальные учебные действия; 2. Познавательные универсальные учебные действия; 3. Коммуникативные универсальные учебные действия. <p><i>Личностные:</i></p> <ul style="list-style-type: none"> - критическое отношение к информации и избирательность её восприятия, - осмысление мотивов своих действий при выполнении заданий, - развитие любознательности, сообразительности при выполнении разнообразных заданий проблемного и эвристического характера, - развитие внимательности, настойчивости, целеустремленности, умения преодолевать трудности, - развитие самостоятельности суждений, независимости и нестандартности мышления, - освоение социальных норм, правил поведения, ролей и форм социальной жизни в группах и сообществах, - формирование коммуникативной компетентности в общении и сотрудничестве с другими обучающимися.
<p>Социальный эффект</p>	<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>
<p>Год разработки</p>	<p>2020 - 2021</p>
<p>Год последней редакции</p>	<p>2021 - 2022</p>

Оглавление

Раздел №1 «Комплекс основных характеристик программы»	7
1.1 Пояснительная записка.	7
1.2 Цель и задачи	10
1.3 Планируемые результаты	10
1.4 Содержание программы	14
Раздел №2 «Комплекс организационно-педагогических условий»	17
2.1 Условия реализации программы	17
2.2 Анализ результативности реализации программы	17
2.3 Методическое обеспечение программы	18
Глоссарий	20
Список литературы	21
Раздел № 3 Приложения	23
1.1 Тема: «Базовый двумерный платформер.»	23

Раздел №1 «Комплекс основных характеристик программы»

1.1 Пояснительная записка.

Дополнительная общеобразовательная общеразвивающая программа имеет техническую направленность.

Уровень программы: стартовый.

Программа разработано в соответствии с:

- Федеральным законом Российской Федерации от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации»;
- Концепцией развития дополнительного образования детей (Распоряжение правительства РФ от 04.09.2014г. №1726-р);
- Порядком организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам (Приказ Минпросвещения России от 09.11.2018г. №196);
- Методическими рекомендациями по проектированию дополнительных общеразвивающих программ, включая разноуровневые программы (письмо Минобрнауки России от 18.11.2015г. № 09-3242);
- Примерными требованиями к программам дополнительного образования детей (Письмо Департамента молодежной политики, воспитания и социальной поддержки детей Минобрнауки России от 11.12.2006 № 06-1844);
- СанПиН 2.4.4.3172-14 «Санитарно-эпидемиологические требования к устройству, содержанию и организации режима работы образовательных организаций дополнительного образования детей», утвержденного постановлением Главного государственного санитарного врача РФ от 04.07.2014 г. №41;
- Программой развития ФГБОУ «ВДЦ Океан» на 2014-2020 г.г., утвержденной распоряжением Правительства РФ от 16.12.2014г. № 2539-р (далее – Программа развития);
- Уставом ФГБОУ ВДЦ «Океан».

– Положением о дополнительной общеобразовательной общеразвивающей программе ФГБОУ ВДЦ «Океан».

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

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

Актуальность программы

В наше время, в связи большим развитием технологий в сфере IT, появился большой спрос на специалистов в различных сферах, начиная от системных администраторов и заканчивая программистами. Профессия программиста достаточно сложна в освоении и, как правило, отталкивает будущих специалистов еще на начальном уровне обучения. Но преодоление первоначального страха перед трудностями дает понимание перспективности и возможностей каждого в изучении языков программирования. Это делает дополнительные образовательные программы в области IT и цифрового творчества безусловно актуальными как для обучающихся, планирующих свою профессиональную траекторию в области информационных технологий, так и для всех остальных.

Наряду с развитием технологий претерпевает изменения и игровая индустрия. Если 30 лет назад человеку было интересно и увлекательно запускать «Arkanoid» (видеоигра для игровых автоматов), то уже сегодня мы можем не только не ходить в специализированные клубы, а находиться дома и лицезреть такие труды как The Elder Scrolls, Warcraft и многие другие. Современные игры — это не только игра одного человека, но и игры в сетях, где люди собираются и могут общаться делиться опытом и т.п. Для игровой индустрии требуются грамотные специалисты, готовые к творческому поиску, быстрым изменениям.

Общие сведения

Данная программа направлена на развитие интереса детей к инженерно-техническим и информационным технологиям, научно-исследовательской и конструкторской деятельности с целью последующего наращивания кадрового

потенциала в высокотехнологичных и наукоемких отраслях промышленности. Обучение по программе способствует развитию технических и творческих способностей, становлению логического мышления, развитию аналитических и конструкторских навыков и компетенций. Междисциплинарность программы также дает возможность углубленного изучения таких предметов как физика, математика, черчение и информатика. Предусмотренные образовательные активности позволяют тренировать креативность, логику, целеустремленность, умение общаться со сверстниками и взрослыми.

Программа ориентирована на развитие первоначальных навыков программирования, дизайна, базового знания построения и концепции игры; на освоение основ создания базовой игровой механики, образов игровых персонажей, планирования игрового процесса на протяжении всей игры, корректировки концепции игры в процессе разработки, координации задействованных в работе специалистов, работы в команде.

Игровой дизайн — это искусство применения дизайна и эстетики в создании игры для получения опыта, в образовательных, физических или экспериментальных целях.

Обучающиеся смогут попробовать себя не только в роли программистов, но и дизайнеров, режиссеров, операторов и сценаристов. Работая в команде, они делят одну задачу на всех, подстраиваются под возможности каждого, привыкают к взаимовыручке.

Отличительные особенности программы заключаются в комплексном подходе к освоению технологии game development (разработки игр), сочетающем в себе развитие геймдизайна, программирования, дизайна, командной работы, проектной деятельности.

Категория обучающихся (адресат программы) – школьники 14-17 лет, проявляющие интерес к технологиям разработки игр, обладающие базовым уровнем знаний по программированию.

Принцип отбора – свобода выбора. Количество учащихся в группе: до 10 человек (ориентировано на количество рабочих мест, из расчета 1 компьютер = 1 ученик), состав может быть разновозрастной и разнополый.

Сроки реализации программы, режим занятий и формы.

Объем – 12 часов. Срок реализации – 1 смена (21 день). Продолжительность занятия – 2 академических часа. Периодичность – 2-3 раза в неделю. В очной форме с включением индивидуальных и групповых форматов, теоретических (лекции) и практических блоков (решение технических задач, техническое творчество и проектирование).

1.2 Цель и задачи

Цель программы:

Цель: развитие инженерного мышления, творческих и проектных компетенций в процессе разработки игр на базе Unity в условиях лаборатории технического творчества.

Задачи программы:

1. Создать условия для освоения обучающимися базовых технических и специальных инженерных компетенций в процессе разработки игр.
2. Содействовать развитию творческих и проектных компетенций в разноформатных активностях технического и цифрового творчества.
3. Организовать пространство коммуникации, способствующее расширению опыта взаимодействия для решения инженерно-технических задач и саморазвития.

1.3 Планируемые результаты

Образовательными продуктами (внешними) станут технические идеи и прототипы игр.

Образовательными результатами (внутренними) будут:

Предметные

Базовые технические компетенции, проявляющиеся в знании:

- основ разработки, тестирования, внедрения и адаптации программного обеспечения,
- инструментов и порядка их применения,

- алгоритмов, алгоритмических конструкций,
 - техники безопасности,
- и умения эти знания применять.

Специальные инженерные компетенции, выражающиеся *в знаниях*:

- особенностей устройства и использования специализированного оборудования,
- основной терминологии в области алгоритмизации и программирования,
- особенностей дизайна, верстки и разработки приложений,
- особенности тестирования и адаптации программного обеспечения для различных платформ,
- принципов программирования на Unity,
- использования различных библиотек;

в умениях:

- составлять алгоритмы для решения прикладных задач,
- реализовывать алгоритмы на компьютере в виде программ,
- отлаживать и тестировать программы,
- разрабатывать приложения самостоятельно в команде соблюдая все технологические требования к дизайну, размещению, функциональным характеристикам,
- сопровождать и адаптировать готовые приложения для внедрения в деятельность конкретного потребителя.

Метапредметные

1. Регулятивные универсальные учебные действия:

- умение принимать и сохранять учебную задачу,
- умение планировать последовательность шагов алгоритма для достижения цели,
- умение ставить цель (создание творческой работы), планировать достижение этой цели,
- умение осуществлять итоговый и пошаговый контроль по результату,

- способность адекватно воспринимать оценку наставника и других обучающихся,

- умение различать способ и результат действия,

- умение вносить коррективы в действия в случае расхождения результата решения задачи на основе её оценки и учёта характера сделанных ошибок,

- умение в сотрудничестве ставить новые учебные задачи,

- способность проявлять познавательную инициативу в учебном сотрудничестве,

- умение осваивать способы решения проблем творческого характера в жизненных ситуациях,

- умение оценивать получающийся творческий продукт и соотносить его с изначальным замыслом, выполнять по необходимости коррекции либо продукта, либо замысла.

2. Познавательные универсальные учебные действия:

- умение осуществлять поиск информации в индивидуальных информационных архивах учащегося, информационной среде образовательного учреждения, федеральных хранилищах информационных образовательных ресурсов,

- умение использовать средства информационных и коммуникационных технологий для решения коммуникативных, познавательных и творческих задач,

- умение ориентироваться в разнообразии способов решения задач,

- умение осуществлять анализ объектов с выделением существенных и несущественных признаков,

- умение проводить сравнение, классификацию по заданным критериям,

- умение строить логические рассуждения в форме связи простых суждений об объекте,

- умение устанавливать аналогии, причинно-следственные связи,

- умение моделировать, преобразовывать объект из чувственной формы в модель, где выделены существенные характеристики объекта (пространственно-графическая или знаково-символическая),

- умение синтезировать, составлять целое из частей, в том числе самостоятельное достраивание с восполнением недостающих компонентов.

3. Коммуникативные универсальные учебные действия:

- умение аргументировать свою точку зрения на выбор оснований и критериев при выделении признаков, сравнении и классификации объектов,
- умение выслушивать собеседника и вести диалог,
- способность признавать возможность существования различных точек зрения и права каждого иметь свою,
- умение планировать учебное сотрудничество с наставником и другими обучающимися: определять цели, функции участников, способы взаимодействия,
- умение осуществлять постановку вопросов: инициативное сотрудничество в поиске и сборе информации,
- умение разрешать конфликты: выявление, идентификация проблемы, поиск и оценка альтернативных способов разрешения конфликта, принятие решения и его реализация,
- умение с достаточной полнотой и точностью выражать свои мысли в соответствии с задачами и условиями коммуникации,
- владение монологической и диалогической формами речи.

Личностные

- критическое отношение к информации и избирательность её восприятия,
- осмысление мотивов своих действий при выполнении заданий,
- развитие любознательности, сообразительности при выполнении разнообразных заданий проблемного и эвристического характера,
- развитие внимательности, настойчивости, целеустремленности, умения преодолевать трудности,
- развитие самостоятельности суждений, независимости и нестандартности мышления,
- освоение социальных норм, правил поведения, ролей и форм социальной жизни в группах и сообществах,

- формирование коммуникативной компетентности в общении и сотрудничестве с другими обучающимися.

1.4 Содержание программы

Таблица № 1 Учебный план

№	Тема	Кол-во часов			Форма аттестации/ контроля
		всего	теория	практика	
1	Введение в курс «Game_Dev» Интерфейс. Возможности .Asset Store Unity.	2	1,5	0,5	Опрос. Вводное анкетирование. Включенное наблюдение
2	Основы алгоритмизации. Подготовка к созданию проекта.	2	1,5	0,5	Результаты заданий, самодиагностика, диагностика валидными методами
3	Создание 2D уровня игры. При помощи TileMaps	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	Разработанные и защищенные проекты, рефлексия.
6	Завершение разработки собственной игр. Подведение итогов курса	2	-	2	
	Всего	12	4	8	

Содержание учебного плана

Занятие № 1

Тема: «Введение в курс «Game_Dev». Интерфейс. Возможности. Asset Store Unity»

Цель: Формирование мотивации учащихся к успешному освоению программы «Game_Dev». 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.

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

Занятие № 2

Тема: «Основы алгоритмизации.»

Цель: Обеспечить углубление, обобщение и систематизацию знаний по алгоритмизации; способствовать формированию умения рационально подбирать методы разработки алгоритмов. Способствовать развитию аналитического и

логического мышления, памяти, внимания, коммуникативных навыков участников программы;

Теория: Изучение терминов и определений: алгоритм, исполнитель алгоритма, среда исполнителя, СКИ (система команд исполнителя), способы описания алгоритма, виды алгоритма.

Практика: Изучение и создание различных блок-схем, разработка алгоритма действия элементарной программы и описание его на «русском языке»

Занятие № 3

Тема: «Создание 2D уровня игры. Создание Scene в Unity.»

Цель: Изучение сути создания уровня при помощи Tile Maps. Physics в Unity и как она используется.

Теория: Объяснение функциональных характеристик и назначение физики. Создания и применение.

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

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, устная рефлексия.

Раздел №2 «Комплекс организационно-педагогических условий»

2.1 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.

2.2 Анализ результативности реализации программы

Одним из показателей результативности является сохранность контингента, 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:

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

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.

2.3 Методическое обеспечение программы

Дидактические и методические материалы

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.

Глоссарий

1. Game_Dev. – от англ. сокращение от *Game Development* – разработка игры, на игровом движке *Unity*
2. Unity – межплатформенная среда разработки игр.
3. Алгоритмизация – процесс составления алгоритмов для решения поставленных прикладных задач.
4. Алгоритм – это точный набор инструкций, описывающих порядок действий некоторого исполнителя для достижения результата, решения некоторой задачи за конечное число шагов.
5. Asset Store – это растущая библиотека ассетов.
6. Ассет или игровой ресурс - цифровой объект, преимущественно состоящий из однотипных данных, неделимая сущность, которая представляет часть игрового контента и обладает некими свойствами.
7. Prefab – Это особый тип ассетов, позволяющий хранить весь *GameObject* со всеми компонентами и значениями свойств.
8. *GameObject* – Любой объект находящийся на сцене игры.
9. Открытое программное обеспечение – распространяющееся бесплатно.
10. Рефлексия – это обращение внимания субъекта на самого себя и на своё сознание, в частности, на продукты собственной активности, а также какое-либо их переосмысление.
11. Система команд исполнителя – это все команды, которые исполнитель умеет выполнять.
12. Среда исполнителя – обстановка, в которой функционирует исполнитель.
13. Технологический уклад – совокупность сопряжённых производств, имеющих единый технический уровень и развивающихся синхронно/
14. 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.

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

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>

<http://immersivededucation.org/>

<http://immersivrededucation.com/>

<https://medium.com/futurepi/a-vision-for-education-and-its-immersive-a-i-driven-future-b5a9d34ce26d>

https://www.ted.com/talks/michael_bodekaer_this_virtual_lab_will_revolutionize_science_class/footnotes?referrer=playlist-10_years_of_ted_talks#t-669397

<https://www.mos.ru/news/item/30181073/>

http://store.steampowered.com/app/482390/The_Night_Cafe_A_VR_Tribute_to_Vincent_Van_Gogh/

http://store.steampowered.com/app/638920/BeanVRThe_Social_VR_APP/

http://store.steampowered.com/app/407060/AltspaceVRThe_Social_VR_App/

<https://www.facebook.com/spaces>

<http://www.xvivo.net/cellscape-vr-biology/>

http://store.steampowered.com/app/451980/The_Body_VR_Journey_Inside_a_Cell

/

<https://melscience.com/vr/>

<https://www.schellgames.com/games/superchem-vr>

Раздел № 3 Приложения

1.1 Тема: «Базовый двумерный платформер.»

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 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;

→ 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 скриптC# (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 старт 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 комплектно 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 воспроизводиться 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 пр. кнопкой мыши и выберите команду Make Transition. Повторить и в обратном направлении.

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

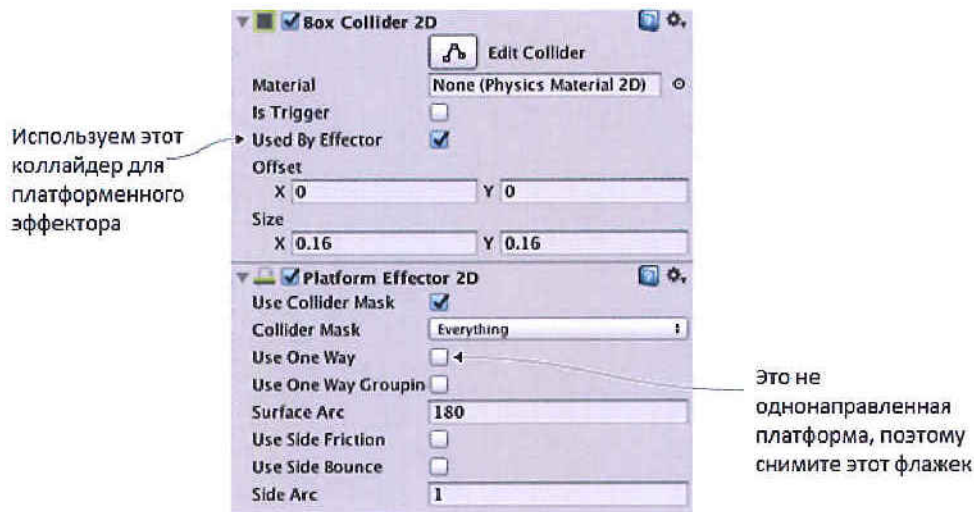
Педагог: Далее открываем скрипт персонажа и добавляем пару строчек в скрипт

```
...  
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 чутка 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?

