

АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОРГАНИЗАЦИЯ
ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
«Ес Программирование»

«Утверждаю»
Директор АНО ДО
«Ес Программирование»
 И.В. Полеников
« 01 » сентябрь 2024г



**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
ПРОГРАММА – ДОПОЛНИТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА
ПО ПРОГРАММИРОВАНИЮ ДЛЯ ДЕТЕЙ
ШКОЛЬНОГО ВОЗРАСТА**

**«Основы программирования и создания игр в Unity»
(11-16 ЛЕТ)**

Екатеринбург – 2024

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Дополнительная общеобразовательная программа «Основы программирования и создания игр в Unity» разработана на основе следующих нормативно-правовых документов:

1. Федеральный Закон «Об образовании в Российской Федерации» от 29.12.2012 № 273-ФЗ.
2. Концепция развития дополнительного образования детей (утверждена распоряжением Правительства РФ от 31 марта 2022 г. № 678-р).
3. Постановлением Главного государственного санитарного врача РФ от 28.09.2020 года №28 «Об утверждении санитарных правил СП 2.4. 3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи» (зарегистрированного от 18.12.2020 №61573).
4. Письмо Минобрнауки России от 18.11.2015 № 09-3242 «О направлении информации» (вместе с «Методическими рекомендациями по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы»).

Направленность дополнительной общеобразовательной программы – *техническое*

Актуальность программы:

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.

Отличительные особенности программы:

Ключевой особенностью курса является его направленность на формирование у учащихся навыков поиска собственного решения поставленной задачи, составления алгоритма решения и его реализации с помощью средств программирования.

Целью программы является обучение детей возраста от 11 до 16 лет созданию трехмерных игр с помощью ПО Unity3D, основам программирования на ЯП C#, и основам 3d моделирования, через создание творческих и прикладных проектов.

Задачи:

Обучающие:

- Освоение основ алгоритмизации и программирования;
- Формирование представления об объектах в объектно-ориентированном программировании;
- Изучение способов обработки событий;
- Освоение способов тестирования программного обеспечения;
- Знакомство со структурой и ведением проекта, умение его представлять.

Развивающие:

- Развитие понимания необходимости обладать достаточными навыками владения информационными технологиями;
- Развитие стремления полностью реализовывать свои проекты;
- Развитие самостоятельности;
- Развитие внимательности;
- Развитие логического и творческого мышления.

Воспитывающие:

- Умение создавать коллективный продукт, помогать окружающим;

- Развитие коммуникативных навыков;
- Умение преодолевать трудности на пути достижения цели;
- Способствовать развитию художественного вкуса;
- Развитие внимательности;
- Сформировать умение формулировать, аргументировать и отстаивать своё мнение.
- Сформировать умение выступать публично с докладами и презентациями.
- Воспитывать аккуратность и дисциплинированность при выполнении работы.
- Способствовать формированию положительной мотивации к трудовой деятельности.
- Способствовать формированию опыта совместного и индивидуального творчества при выполнении командных заданий.

Программа адресована для детей от 11 лет до 16 лет, индивидуально, в группе или онлайн обучение. Рекомендуемая частота занятий – 1 раз в неделю по 2 академических часа.

Формы организации образовательного процесса

- индивидуальная
- групповая
- онлайн

Виды занятий:

- теоретические
- практические

Ожидаемые результаты и способы определения их результативности:

Знать:

- отдельные способы планирования деятельности;
- составление плана предстоящего проекта в виде рисунка, схемы;
- базовый синтаксис языка программирования C#.
- устройство игровых объектов в Unity
- способы создания игр

Уметь:

- составить план проекта, включая: выбор темы;
- анализ предметной области;
- разбиение задачи на под-задачи;
- проанализировать результат и сделать выводы;
- найти и исправить ошибки;
- подготовить небольшой отчет о работе;
- публично выступить с докладом;
- наметить дальнейшие пути развития проекта;
- реализовывать простые алгоритмы с помощью языка программирования C#;
- создавать и внедрять 3d модели в проекты Unity;
- создавать скомпилированную версию игры.

иметь первичные навыки:

- работы в группе;
- ведения дискуссии;
- донесения своих мыслей до других.

Ожидаемые результаты обучения по программе

Учащиеся овладевают следующими знаниями, умениями и способами деятельности:

- знают принципы и структуру Unity проектов, формы представления и управления информацией в проектах;
- умеют спроектировать, изготовить и запустить проект в Unity;
- владеют способами работы с изученными программами;
- знают и умеют применять при создании Unity проектов основы организации кода.
- способны осуществлять рефлексивную деятельность, оценивать свои результаты, корректировать дальнейшую деятельность по разработке Unity проектов.

Программа рассчитана на 68 часов (2 часа в неделю). Занятия разделены на академические часы (40 минут) с перерывами между ними по 10 минут. Обучение проходит индивидуально, в группе или в формате онлайн обучения.

Форма подведения итогов реализации программы:

Защита творческих работ, созданных в среде разработки Unity. Представление проектов через презентацию перед преподавателем и другими студентами.

Учебный календарный график

№	Месяц	Форма занятия	Кол-во часов	Раздел, тема занятия	Форма контроля
Раздел I. Основы 3D моделирования					
1	Сентябрь	Учебное занятие	2	Введение в курс. Инструктаж по ТБ. Знакомство с MagicaVoxel	Тестирование, практическое задание
2	Сентябрь	Учебное занятие	2	Основы проектирования 3D моделей	Тестирование, практическое задание
3	Октябрь	Учебное занятие	2	Моделирование персонажа	Практическое задание
Раздел II. Анимация через Mixamo					
4	Октябрь	Учебное занятие	2	Знакомство с сервисом Mixamo. Импорт модели	Практическое задание, наблюдение
5	Октябрь	Учебное занятие	2	Настройка скелета модели и анимаций	Тестирование, практическое задание
6	Октябрь	Учебное занятие	1	Сбор необходимых анимаций для проекта	Практическое задание
Раздел III. Введение в Unity					
7	Ноябрь	Учебное занятие	2	Знакомство с интерфейсом Unity и инструментами позиционирования	Тестирование, практическое задание
8	Ноябрь	Учебное занятие	2	Импорт и настройка .fbx моделей	Практическое задание
Раздел IV. Настройка анимации в Unity					
9	Ноябрь	Учебное занятие	2	Подготовка модели и создание Animator Controller	Тестирование, практическое задание
10	Ноябрь	Учебное занятие	2	Подключение Animator Controller к игровому объекту	Практическое задание
Раздел V. Основы C#					
11	Декабрь	Учебное занятие	2	Введение в язык программирования C#. Переменные и методы	Тестирование, практическое задание
12	Декабрь	Учебное занятие	2	Классы и модификаторы доступа	Наблюдение, практическое задание
Раздел VI. Создание скриптов в Unity					

13	Январь	Учебное занятие	2	Создание и прикрепление скрипта движения персонажа	Практическое задание
14	Январь	Учебное занятие	2	Создание скрипта управления анимацией	Практическое задание, тестирование
Раздел VII. Создание и настройка интерактивного окружения					
15	Февраль	Учебное занятие	2	Виды интерактивного окружения и обработка событий касания	Тестирование, практическое задание
16	Февраль	Учебное занятие	2	Префабы и их использование для игровых объектов	Практическое задание
Раздел VIII. Графический интерфейс пользователя					
17	Март	Учебное занятие	2	Canvas и программирование интерфейса через UnityEngine.UI	Тестирование, практическое задание
18	Март	Учебное занятие	2	Создание стартового меню	Практическое задание
Раздел IX. Сборка и тестирование игры					
19	Апрель	Учебное занятие	2	Проектирование уровней и сцен	Практическое задание
20	Апрель	Учебное занятие	2	Сборка игры и создание финальных файлов	Презентация проекта, тестирование
Итоговая аттестация					
21	Май	Учебное занятие	1	Защита проекта. Презентация результатов	Презентация, защита проекта

Учебный план

п/п	Название раздела, темы	Всего	Теория	Практика	Форма аттестации/контроля
1	Основы 3D моделирования				
1.1	Введение в 3D модели	2	1	1	Практическое задание, наблюдение
1.2	Знакомство с MagicaVoxel: Создание простых моделей	2	1	1	Практическое задание
1.3	Основы проектирования гуманоидных 3D моделей	2	1	1	Практическое задание
1.4	Моделирование собственного персонажа	2	1	1	Практическое задание, наблюдение
2	Анимация через Mixamo				
2.1	Анимация через Mixamo: Знакомство с сервисом	2	1	1	Практическое задание
2.2	Импорт модели и настройка скелета	2	1	1	Практическое задание
2.3	Сбор необходимых анимаций	2	1	1	Практическое задание
3	Введение в Unity				
3.1	Введение в Unity: Знакомство с интерфейсом	2	1	1	Практическое задание, тестирование 1
3.2	Изучение инструментов позиционирования	2	1	1	Практическое задание
4	Настройка анимации в Unity				
4.1	Импорт и начальная настройка .fbx моделей	2	1	1	Практическое задание
4.2	Настройка анимации в Unity	2	1	1	Практическое задание
4.3	Подготовка модели и файлов анимации	2	1	1	Практическое задание
5	Основы C#				
5.1	Создание Animator Controller	2	1	1	Практическое задание
5.2	Составление дерева анимации	2	1	1	Практическое задание
5.3	Подключение Animator Controller к игровому объекту	2	1	1	Практическое задание
5.4	Введение в C#: основы программирования	2	1	1	Практическое задание, наблюдение
5.5	Переменные и способы их определения	2	1	1	Практическое задание
5.6	Методы и способы их определения	2	1	1	Практическое задание
5.7	Модификаторы доступа и типы данных	2	1	1	Практическое задание
5.8	Классы и способы их определения	2	1	1	Практическое задание, тестирование 2
6	Создание скрипта движения персонажа				
6.1	Создание скрипта движения персонажа	2	1	1	Практическое задание
6.2	Инструменты создания и редактирования C# скриптов в Unity	2	1	1	Практическое задание
6.3	Создание и прикрепление скрипта движения	2	1	1	Практическое задание
6.4	Создание скрипта управления анимацией	2	1	1	Практическое задание
7	Создание и настройка интерактивного окружения				
7.1	Изучение видов интерактивного окружения в играх	2	1	1	Практическое задание

7.2	Обработка событий касания	2	1	1	Практическое задание
7.3	Префабы, как способы дублирования игровых объектов	2	1	1	Практическое задание
8	Графический интерфейс пользователя				
8.1	Графический интерфейс: Знакомство с Canvas и инструментами интерфейса	2	1	1	Практическое задание, тестирование 3
8.2	Библиотека UnityEngine.UI для программирования интерфейса	2	1	1	Практическое задание
8.3	Создание стартового меню	2	1	1	Практическое задание
9	Сборка игры				
9.1	Проектирование и создание уровней с помощью сцен	2	1	1	Практическое задание
9.2	Билд игры в набор конечных файлов	2	1	1	Практическое задание
9.3	Финальная работа: проверка проекта и отладка	2	1	1	Практическое задание
9.4	Презентация проектов: представление итоговой игры	2	1	1	Презентация

Содержание изучаемого курса:

1. Основы 3D моделирования

Теория: Знакомство с 3D редактором MagicaVoxel и изучение правил моделирования Humanoid моделей.

Практика: Создание персонажа для будущей игры.

2. Анимация через Mixamo

Теория: Импорт, настройка и экспорт анимации с помощью сервиса Mixamo.com

Практика: Создание и загрузка анимации для собственной игры

3. Введение в Unity

Теория: Основы работы в среде трёхмерной разработки Unity/

Практика: Создание композиции из примитивов.

4. Настройка анимации в Unity

Теория: Работа с .fbx моделями в Unity. Настройка аватара.

Практика: Создание и настройка Animator Controller главного персонажа

5. Основы C#

Теория: Введение в язык программирование C#. Переменные и способы их определения, Методы и способы их определения. Модификаторы доступа. Типы данных. Классы и способы их определения.

Практика: Создание тренировочного скрипта с реализацией всех основных структур данных.

6. Создание скрипта движения персонажа

Теория: Инструменты создания и редактирования C# скриптов в Unity. Создание и прикрепление скриптов к игровым объектам Юнити.

Практика: Создание скриптов движения и управления анимацией.

7. Создание и настройка интерактивного окружения

Теория: Обработка событий касания в элементах интерактивного окружения в играх. Префабы, как способы дублирования готовых игровых объектов

Практика: Создание префабов для интерактивного окружения и наполнения сцен.

8. Графический интерфейс пользователя.

Теория: Canvas и инструменты работы с интерфейсом. Библиотека UnityEngine.UI для программирования интерфейса пользователя.

Практика: Создание стартового меню.

9. Сборка игры.

Теория: Проектирование и создание уровней с помощью сцен.

Практика: Билд игры в набор конечных файлов.

Методическое обеспечение дополнительной общеобразовательной программы дополнительного образования детей:

Учебные пособия:

1. Дрейер, М. С# для школьников : школьное пособие / М. Дрейер ; под ред. В. Биллиг. - М. : Интернет-Университет Информационных Технологий, 2010. - 126 с. [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=233057>
2. Джозеф Хокинг Unity в действии. Мультиплатформенная разработка на С#. - СПб: Питер, 2016. - 336 с.

ТЕХНИЧЕСКОЕ ОСНАЩЕНИЕ

- Ученический стол со стульями.
- Стол и стул (для преподавателя).
- Ноутбук (для преподавателя).
- Мышь компьютерная 3шт.
- Ноутбук (для ученика) 2шт.

Ожидаемые результаты и способы определения их результативности

По итогам реализации программы, учащиеся

будут знать:

- принципы создание 3d игр в среде разработки Unity;
- синтаксис языка программирования С#;
- основы 3d моделирования.

По итогам реализации программы, учащиеся

будут уметь:

- создавать скрипты на языке программирования С#;
- создавать, настраивать и позиционировать игровые объекты в Unity;
- запускать и отлаживать проекты в Unity.

Формы определения результативности обучения

Результаты освоения программы отслеживаются по итогам опросов, выполнения практических заданий и тестирования.

Условия реализации программы

Для успешной реализации программы в очном формате необходим ноутбук с установленным на нем любым современным браузером и компьютерная мышь.

Для успешной реализации программы с дистанционным форматом необходим компьютер с доступом к сети Интернет (желательно со скоростью подключения не меньше 50мб/сек), подключенными колонками (или наушниками). На компьютере должен быть установлен любой современный браузер и платформа Discord.

Организационно-педагогические условия

При проведении занятия с использованием дистанционных образовательных технологий, электронного обучения, в водной его части следует обозначить правила работы и взаимодействия (объяснить учащимся технические особенности работы и правила обмена информацией). В процессе занятия педагогу необходимо четко давать инструкции выполнения заданий.

Список литературы или Интернет источники

1. Доусон М. Програмируем на Python. – СПб.: Питер, 2014 г. – 416 с.
2. Официальный сайт программы [Электронный ресурс] – Режим доступа: <https://docs.python.org/>, свободный.
3. Сайт, среда разработки для языка Python. [Электронный ресурс] – Режим доступа: <https://www.jetbrains.com/pycharm/?fromMenu>, свободный.

Тест №1. По курсу Unity

Имя студента	
--------------	--

1. Зажав какую кнопку, можно двигаться по сцене с помощью клавиш WASDEQ?

- ПКМ
- ЛКМ
- Esc
- Ctrl

2. Какой компонент добавляет цвет к игровому объекту в Unity?

- Rigidbody
- Collider
- Material
- Join Point

3. На каком языке программирования создаются скрипты в Unity?

- Python
- JavaScript
- Java#
- C#

4. Что добавляется к объекту, чтобы распознать, что стенки коснулся именно он?

- Скрипт
- Компонент
- Тэг
- Материал

5. Какое сочетание клавиш позволяет дублировать объекты?

- Ctrl+S
- Ctrl+A
- Ctrl+D
- Ctrl+B

6. Что такое Voxel?

- Круг в графике
- Шар в 3D-графике
- Квадрат
- Куб в 3D-графике

7. Какой инструмент в MagicaVoxel позволяет удалять воксели?

- Paint
- Erase
- Attach
- Fit

8. Какая кнопка в MagicaVoxel максимально сжимает пустое пространство вокруг модели?

- Paint
- Erase
- Attach
- Fit

9. Как называется сайт-приложение от Adobe для создания анимации

- Unity
- Magica
- Mixamo
- Voxel

10. Какой тип Модели нужно использовать для работы с человекоподобной анимацией?

- Generic
- Humanoid
- Legacy

11. Какой Unity-компонент делает игровой объект твёрдым, но не добавляет ему физики?

- Collider
- Animator
- Rigidbody
- Mesh Renderer

12. Какой Unity-компонент придаёт игровому объекту физических свойств?

- Collider
- Animator
- Rigidbody
- Mesh Renderer

13. Как называется раздел Unity, который позволяет смешивать несколько видов анимации?

- Material
- Animator Controller
- Folder
- Script

14. Какая команда позволяет создать «дерево Анимации»

- From new blend tree
- Create empty
- Add parameter
- Float

15. Параметры какого типа могут содержать дробные числа?

- Trigger
- Bool
- Float
- Int

Тест №2. По курсу Unity. Основы C#

Имя студента	
--------------	--

1. Что такое переменная?

- Переменная – это значение результата работы программы
- Переменная – часть кода, в которой указывается смысл программы
- Переменная – название для операторов в коде
- Переменная – ячейка памяти в компьютере, в которой хранится значение

2. Какой тип данных обозначает целое число?

- int
- bool
- float
- string

3. Какой тип данных обозначает логическую переменную?

- int
- bool
- float
- string

4. Выберите специальный символ, который обозначает комментарий в C#?

- *
- \
- //
- >

5. Что такое метод?

- Метод – это контейнер для набора инструкций, которые выполняются каждый раз, когда его вызываешь
- Метод – это хранилище переменных, чтобы было их удобно использовать
- Метод - скрипты для игр
- Метод - механизм обработки игровых объектов

6. С помощью какого метода мы получаем доступ к Unity-компоненту?

- Update
- Start
- .GetAxis
- GetComponent

7. С помощью какого класса мы получаем ввод данных от пользователя?

- Input
- MonoBehaviour
- Transform
- GameObject

8. Выберите правильное название переменной??

- myName
- "my_name"
- 13MyName
- ?Name
- my name
- my/name

9. Что будет выведено в консоль?

Примечание: `Debug.Log()` - выводит информацию в консоль

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class Test1 : MonoBehaviour
6 {
7     public int number = 15;
8     // Start is called before the first frame update
9     void Start()
10    {
11        Debug.Log(number + 15);
12    }
13
14    // Update is called once per frame
15    void Update()
16    {
17    }
18 }
19
```

10. Выберите математический оператор, который обозначает умножение?

- /
- +
- *
- x

11. Что будет выведено на экран?

Примечание: `Debug.Log()` - выводит информацию в консоль

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class Test1 : MonoBehaviour
6 {
7     public string word = "Я люблю ";
8     public string name = "Unity ";
9
10    // Start is called before the first frame update
11    void Start()
12    {
13        Debug.Log(word + name + 2021);
14    }
15
16    // Update is called once per frame
17    void Update()
18    {
19    }
20 }
```

- word name 2021
- Я люблю юнити 2021
- Я люблю 2021
- Я люблю Unity 2021
- Unity 2021

12. Что будет выведено на экран?

Примечание: `Debug.Log()` - выводит информацию в консоль

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class Test1 : MonoBehaviour
6 {
7     public float speed = 5f;
8     public float jump = 15f;
9     public string name = "Player"
10
11    // Start is called before the first frame update
12    void Start()
13    {
14        Debug.Log("Меня зовут " + name);
15        Debug.Log("Моя скорость " + speed);
16        Debug.Log("Моя сила прыжка " + jump);
17    }
18
19    // Update is called once per frame
20    void Update()
21    {
22    }
23 }
24 }
```

Тест №3. По курсу Unity.

Имя студента	
--------------	--

- 1. С помощью какой механики в уроке определяется соприкосновение персонажа с землёй?**
 - Collision
 - Trigger
 - Raycast
 - SphereOverlap

- 2. С помощью какого ключевого слова создаётся корутина?**
 - void
 - public
 - private
 - IEnumerator

- 3. Как называется Unity-Axis с помощью которой можно считать движение мыши влево-вправо?**
 - Horizontal
 - Vertical
 - Mouse X
 - Mouse Y

- 4. Как называется Unity-Axis с помощью которой можно считать движение мыши вверх-вниз?**
 - Horizontal
 - Vertical
 - Mouse X
 - Mouse Y

5. Как называется метод класса Transform, который позволяет вращать объект?

- transform.Translate()
- transform.forward
- transform.right
- transform.Rotate()

6. Какое свойство компонента Box Collider, дает возможность проходить объект насквозь, но при это считывать соприкосновения?

- Mass
- X
- Y
- IsTrigger

7. Какой метод вызывается автоматически, когда другой коллайдер входит в коллайдер с isTrigger?

- OnCollisionExit
- OnCollisionEnter
- Update
- OnTriggerEnter

8. Шаблон объект, со всеми компонентами и значениями, сохранённый в папке проекта называется:

- Коллайдер
- Префаб
- Контроллер
- Темплейт

9. Как называется метод, который уничтожает объекты на сцене?

- LookAt
- Invoke
- SetFloat
- Destroy

10. Что делать, если необходимо создать какое-либо действие?

- Использовать переменные
- Использовать игровой объект
- Использовать метод
- Использовать модификатор доступа

11. Что будет выведено на экран?

Примечание: `Debug.Log()` - выводит информацию в консоль?

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class Test1 : MonoBehaviour
6  {
7      public int number1 = 3;
8      public int number2 = 5;
9
10     // Start is called before the first frame update
11     void Start()
12     {
13         ComputeNumber();
14     }
15
16     void ComputeNumber()
17     {
18         Debug.Log(number1 + number2);
19     }
20 }
```

- 35
- 8
- number1 + number2
- Ничего не произойдёт

12. Что такое объявление метода?

- Это его создание, присвоение ему имени и определение команд, которые он будет выполнять
- Это его вызов
- Это разрешение доступа на использование переменных
- Это создание скрипта

13. Что такое вызов метода?

- Создание метода
- Использование метода
- Добавление метода к игровому объекту

14. Какими символами можно выделить тело метода?

- ()
- []
- {}
- <>

15. Выберите правильные написание вызовов методов?

- PrintNumer();
- PrintNumer;
- "PrintNumer()";
- printNumer;
- PrintNumer()?;

16. В каком варианте правильно объявлен метод?

```
16 void Example(){  
17     Debug.Log("hello world")
```

```
16 void Exmaple()  
17     Debug.Log("hello world"){}
```

```
16 void Exmaple()  
17     {  
18         Debug.Log("hello world")  
19     }
```

```
16 void { Exmaple()  
17     Debug.Log("hello world")  
18 }
```