Приветствую, будущий маг кода и повелитель виртуальных миров! Я буду твоим гидом в захватывающем мире C# и Unity.

Забудь скучные лекции и сухие примеры вроде Calculator.cs. Мы будем учиться, создавая игры. Каждая новая тема — это новый кирпичик в твоей собственной игровой вселенной. К концу курса у тебя будет не только багаж знаний, но и портфолио из маленьких, но настоящих игровых проектов.

Курс: "C# Магия: От нуля до героя Unity"

Девиз: *Пиши код. Создавай миры. Побеждай.*

Наш арсенал:

- **Язык:** C# 10 (.NET 6+)
- **IDE:** Microsoft Visual Studio 2022 (с тёмной темой, ведь все крутые ребята так делают!)
- **Игровой движок:** Unity 2022 LTS (начиная с 6-го модуля)

Программа курса: Наше эпическое приключение

Модуль 0: Пролог — Установка и Первое заклинание (1 занятие)

- **Квест:** "Инициализация героя"
- **Задача:** Установить и настроить всё необходимое ПО.
- **Геймификация:** Создадим своего первого цифрового питомца — консольное приложение, которое будет представлять тебя как разработчика.
- **Содержание:**
  - Установка Visual Studio 2022 и необходимых компонентов (.NET, игры).
  - Знакомство с интерфейсом IDE.
  - Первая программа: `Console.WriteLine("Hello, World! Я — маг C#!");`
  - Понятие `Main`-метода как точки входа в нашу вселенную.

Модуль 1: Основы алхимии — Синтаксис и типы данных (4 занятия)

- **Квест:** "Зельеварение для начинающих"
- **Задача:** Понять фундаментальные строительные блоки кода.
- **Геймификация:** Мы создадим текстовый симулятор варки зелий.

    - Игрок вводит названия ингредиентов (строки), их количество (числа), определяет, является ли ингредиент редким (булево значение).
    - Программа вычисляет стоимость зелья, шанс успеха и выдаёт результат.
- **Содержание:**
    - **Переменные и типы данных:** int, float, double, bool, char, string. Это наши ингредиенты.
    - **Операторы:** Арифметические (+, -, *, /), сравнения (>, ==, !=), логические (&&, ||, !). Это наши магические жесты для смешивания.
    - **Ввод/вывод в консоли:** Console.ReadLine(), Console.WriteLine().
    - **Преобразование типов:** Как превратить "5" (строку) в 5 (число).
    - **Ключевое слово var.** Позволяет компилятору самому угадать тип — как магическое предсказание.

Модуль 2: Управление потоком магии — Условные операторы и циклы (4 занятия)

- **Квест:** "Лабиринт Принятия Решений"
- **Задача:** Научить программу принимать решения и повторять действия.
- **Геймификация:** Создадим текстовый квест-игру "Сокровища Древнего Храма".
    - Игрок выбирает, в какую дверь войти (if/else), какой предмет взять (switch-case).
    - Циклы (for, while) используются для разгадывания загадок (ввести правильный код за 3 попытки) или для прохождения коридора с ловушками.
- **Содержание:**
    - **Условные операторы:** if, else if, else. Ветвление сюжета.
    - **Оператор выбора:** switch-case. Выбор из множества вариантов (например, выбор оружия).
    - **Циклы:** for (для действий с известным количеством повторений), while (пока условие истинно), do-while.

- **Операторы break и continue.** Волшебные телепорты внутри циклов.

Модуль 3: Магия структур — Методы и массивы (4 занятия)

- **Квест:** "Архив Древних Свитков и Сумка Путешественника"
- **Задача:** Научиться структурировать код и данные.
- **Геймификация:** Создадим систему инвентаря и заклинаний для нашего героя.
  - **Массивы** (string[] inventory = new string[5]) — это наш инвентарь, где есть слоты для предметов.
  - **Методы** — это наши заклинания. Например:
    - void CastHeal() — заклинание лечения (просто выводит текст).
    - int CalculateDamage(int weaponPower) — заклинание атаки, которое возвращает урон.
    - void AddToInventory(string item) — метод для добавления предмета в массив-инвентарь.
- **Содержание:**
  - **Методы:** Создание, вызов, возвращаемые значения (void, int, string...), параметры.
  - **Массивы и цикл foreach.** Идеально для перебора всех предметов в инвентаре.
  - **Пространства имён (Namespaces).** Как разные отделы в магической библиотеке.

Модуль 4: Создание существ — Введение в ООП (Объектно-Ориентированное Программирование) (6 занятий)

- **Квест:** "Бестиарий: Создай свою армию"
- **Задача:** Понять основные концепции ООП: классы, объекты, инкапсуляцию, наследование, полиморфизм.
- **Геймификация:** Мы создадим иерархию игровых существ.

- **Классы и Объекты:** Создаём класс Monster. Затем создаём объекты: Monster goblin = new Monster();, Monster dragon = new Monster();.
- **Поля и свойства:** У монстра есть Health, Damage, Name (с использованием инкапсуляции).
- **Методы:** Attack(), TakeDamage().
- **Наследование:** Создаём класс BossMonster : Monster, который наследует всё от обычного монстра, но имеет дополнительную способность SpecialAttack().
- **Полиморфизм:** Переопределяем метод Attack() для BossMonster с помощью override, чтобы его атака была сильнее.

- **Содержание:**
  - **Классы и объекты.** Чертёж (класс) и созданный по нему автомобиль (объект).
  - **Модификаторы доступа:** public, private, protected. Кто имеет доступ к "секретным рецептам" нашего класса.
  - **Конструкторы.** Специальный метод для "рождения" объекта.
  - **Наследование, полиморфизм, виртуальные и переопределяемые методы (virtual, override).**

Модуль 5: Сокровищница данных — Коллекции и LINQ (4 занятия)

- **Квест:** "Великая Библиотека и Магический Каталог"
- **Задача:** Познакомиться с продвинутыми структурами данных и языком запросов LINQ.
- **Геймификация:** Создадим каталог всех заклинаний в игре и систему их поиска/фильтрации.
  - **Списки (List<T>)**: Динамический список заклинаний (в отличие от массива, его размер можно менять).
  - **Словари (Dictionary<TKey, TValue>)**: Коллекция "Ключ-Значение", например, для быстрого поиска заклинания по его названию.
  - **LINQ:** Магические запросы к нашим коллекциям!
    - "Найди все огненные заклинания" (Where(spell => spell.Element == "Fire")).

        - "Отсортируй заклинания по силе урона" (OrderBy(spell => spell.Damage)).
        - "Покажи только названия всех заклинаний" (Select(spell => spell.Name)).
- **Содержание:**
    - **Generic-коллекции:** List<T>, Dictionary<TKey, TValue>.
    - **LINQ синтаксис запросов и методы расширения (Where, Select, OrderBy, FirstOrDefault).**
    - **Лямбда-выражения:** x => x > 5 — наша магия для фильтрации.

Модуль 6: Финальный босс — Интеграция с Unity и создание первой 2D игры (5 занятий)

- **Квест:** "Падение Тёмного Куба"
- **Задача:** Применить все полученные знания в реальном игровом проекте на Unity.
- **Геймификация:** Создадим простой 2D-шутер типа "Space Invaders".
    - Создаём игровой объект (корабль) и привязываем к нему скрипт PlayerController.cs (это наш C# класс!).
    - **Поля:** public float speed; — теперь это можно настраивать прямо в инспекторе Unity!
    - **Методы:** Update() (движение корабля с помощью Input), OnTriggerEnter2D() (столкновение с пулей).
    - Создаём префабы (шаблоны) врагов и пуль, используя знания о классах и объектах.
    - Управляем волнами врагов с помощью List<Enemy>.
    - Используем события и делегаты для системы очков (public static Action<int> OnScoreChanged;).
- **Содержание:**
    - **Основы Unity:** Иерархия, сцена, инспектор, префабы.
    - **Структура MonoBehaviour:** Start(), Update().
    - **Работа с компонентами:** Transform, Rigidbody2D.
    - **Физика:** Коллайдеры и триггеры.
    - **Ввод пользователя:** Input.GetKey().

    - **Создание и уничтожение объектов (Instantiate, Destroy).**

Система мотивации и достижений

- **Achievements (Достижения):**
    - *Первое заклинание:* Запустил первую программу.
    - *Алхимик:* Создал работающий симулятор зелий.
    - *Повелитель циклов:* Прошёл лабиринт без единой ошибки.
    - *Архитектор кода:* Создал иерархию классов с наследованием.
    - *Библиотекарь:* Применил 5 разных LINQ-запросов.
    - *Герой Unity:* Завершил финальный проект.
- **"Уровни" разработчика:** После каждого модуля студент "получает уровень" (Новичок -> Ученик -> Подмастерье -> Маг -> Архимаг).
- **Формат:** Мини-лекция -> Практическое заклинание (кодинг) -> Обсуждение -> Домашнее задание в виде игрового квеста.

Готов стать Архимагом C#? Наше приключение начинается! Вместе мы превратим эти странные символы в код, который оживит твои игровые миры.

Вперёд, кодить!