



Unity Game Developer. Basic

Best Practice по разработке игр и их монетизации (3 проекта на Unity за 5 месяцев)

Длительность курса: 134 академических часа

1 Введение в Unity. Основы программирования и компьютерной графики

1 Знакомство с Unity

Цели занятия:

охарактеризовать экосистему Unity;
установить и настроить Unity-редактор через Unity Hub;
познакомиться с основными окнами редактора, концепцией сцены, ассетов, игровых объектов и компонентов.

Краткое содержание:

что такое Unity и где используется;
элементы экосистемы Unity;
Unity Hub, загрузка и установка Unity-редактора;
окна редактора, сцена;
ассеты, игровые объекты и компоненты.

Домашние задания

1 Знакомство с Unity и работа с ассетами из Asset Store

Цель: Познакомиться с Unity и отработать пайплайн работы с Asset Store и ассетами

Зарегистрировать учетную запись Unity
<https://assetstore.unity.com/>

Выбрать и добавить к себе ассеты. Обратит особое внимание на официальные ассеты (Essentials, Free)

Загрузить и установить Unity через Unity Hub
<https://unity.com/ru/download>

Создать проект и добавить в него выбранные ассеты через Package Manager (фильтр - My Assets)

Изучить содержимое ассетов, иерархию папок, иерархию проектов, компоненты. Запустить сцены - примеры, или добавить ассеты к себе в сцену и запустить.

Написать краткие выводы и прислать ссылки на ассеты, которые удалось посмотреть.

2 Основы компьютерной графики в играх

Цели занятия:

познакомиться с объектами в компьютерной графике; описать что такое mesh и как происходит рендеринг; изучить инструмент Frame Debugger в Unity.

Краткое содержание:

2D и 3D объекты;
глобальные и локальные координаты;
Mesh, текстура, материал;
рендеринг и шейдеры;
Frame Debugger.

Домашние задания

1 Практика работы с материалами и шейдерами

Цель: Создавать новые материалы и выбирать шейдеры.

1. Создайте несколько материалов (5-10).
2. Установите на каждом материале разный шейдер (5 встроенных, 5 найденных в AssetStore).
3. Создайте на сцене объекты к которым применены данные шейдеры.
4. Исследуйте, при изменении каких параметров шейдера, какие визуальные изменения происходят на объекте.

*** Опциональное дополнительное задание:
запустите для любой из ранее загруженных сцен инструмент FrameDebugger. Какую информацию он предоставляет? Для чего может быть полезен?

3 Система контроля версия GIT

Цели занятия:

перечислить особенности использования гита для юнити;
назвать основные команды гита.

Краткое содержание:

гит система;
создание проекта с нуля, добавление поддержки гита;
команды гита.

4 Введение в C#. Часть 1

Цели занятия:

познакомиться с основами программирования на языке C#;
назвать типы данных, что такое переменные и как работать с файлами.

Краткое содержание:

установка, подключение и настройка IDE (Rider, Visual Studio, Visual Studio Code);
типы данных;
константы и переменные;
преобразования типов данных;
операции со строками;
ввод и вывод данных, работа с файлами.

Домашние задания

1 ДЗ по C#: Часть 1

1 Установить IDE

2 Создать переменные на каждый чисельный тип данных

3 Привести пример преобразования из одного типа в другой (4 разные)

4 Создать строку куда записать все оригинальные переменные + их преобразование

5 Создать файл и записать туда все переменные + строка.

* доп задание

6 Создать на все операции кнопки чтобы была возможность запустить и изменить данные в игре

5 Введение в C#. Часть 2

Цели занятия:

познакомиться с основными операторами в языке C#;
обрабатывать исключения;
описать что такое сериализация.

Краткое содержание:

условные операторы;
циклы;
обработка исключений;
массивы;
перечисления;
ключевые слова ref и out;
сериализация.

Домашние задания

1 ДЗ по C#: Часть 2

ДЗ по шарпу: Часть два

1 Создать переменную булл, в зависимости от ее значения создать массив int\float

2 Создать переменную типа int\float. Заполнить массив так, чтобы каждый элемент был квадратом предыдущего.

3 Создать и отловить исключение (переполнение значения)

4 Создать функцию на вход передать переменную (которая в задании 2). После выполнения функции. Сделать пункт 3\4

5 Создать функцию на вход передать переменную REF(которая в задании 2). После выполнения функции. Сделать пункт 3\4

6 Создать функцию на вход передать переменную OUT(которая в задании 2). После выполнения функции. Сделать пункт 3\4

7 Объявить структуру, которая будет содержать все элементы для предыдущих заданий.

8 Записать ее в файл как в первом занятии.

9 Записать ее в файл через сериализацию.

10 Считать данные из файла.

* доп задание

11 Создать кнопки для управлением всем процесом из игры

6 Математика в Unity

Цели занятия:

узнать на каком уровне необходимо знать математику, чтобы разрабатывать игры;
познакомиться с какими математическими конструкциями и понятиями вы точно встретитесь в Unity.

Краткое содержание:

векторы и операции с ними;
матрицы и операции с ними;
преобразования;
кватернионы;
интерполяция;
класс Mathf.

7 ООП. Часть 1

Цели занятия:

познакомиться с основами Объектно-Ориентированного программирования (ООП):
узнать что такое класс, из чего он состоит и как наследуется.

Краткое содержание:

структуры, классы и методы;
идентификаторы доступа;
Anonymous и Nullable типы;
интерфейсы, абстрактные классы;
наследование.

8 ООП. Часть 2

Цели занятия:

познакомиться с еще одним столпом ООП - полиморфизмом, узнать какие бывают коллекции и как с ними работать.

Краткое содержание:

полиморфизм;
коллекции;
делегаты.

Домашние задания

1 ООП

Создать класс Enemy.

Наделить его способностью передвигаться в заданные координаты.

Наделить его способностью атаковать всех в заданной области вокруг себя

Наделить его способностью принимать команды ввода от игрока\компьютера

Наделить его способностью получать урон

Создать класс Warrior, который будет наследовать Enemy

Наделить его способностью перемещаться в заданные координаты и атаковать всех кого он встретит на пути

Создать класс Archer, который будет наследовать Enemy

Наделить его способностью атаковать всех в заданной области в радиусе атаки

Доп задание.

Создать сцену и расставить кубы разного цвета (враги). Дать возможность игроку выбирать куб и применять его способности

2 ДЗ (альтернативное)

Цель: Цель этого ДЗ - повторить и закрепить тему ООП на практике. В нём есть как готовый код с примерами, так и нереализованные классы, которые необходимо дописать.

1. Скачайте и запустите проект в Unity. Он создан в версии 2021.3.5f1, но должен работать и в более старых и более новых.

2. Откройте сцену Homework.

3. Откройте проект в Visual Studio, Rider или другом редакторе кода.

4. Задания описаны в классах Spawner, Dog, SadDog и Run.

9 Базовые алгоритмы

Цели занятия:

узнать как и где применяются алгоритмы в играх, какие виды бывают;
познакомиться с реализацией нескольких классических алгоритмов.

Краткое содержание:

применение алгоритмов в играх;
виды алгоритмов;
сложность алгоритмов;
примеры реализации простых алгоритмов (поиск, сортировка).

Домашние задания

1 №5 Реализация алгоритма

Цель: Написать алгоритм поиска и сортировки, любой кроме описанных на уроке.
Доп задание написать алгоритм поиска пути (можно найти готовый и разобраться в нем)

1 Персонажи и игровой уровень

Цели занятия:

познакомиться с настройкой 3D-уровня из готовых ассетов;
создать префабы;
работать с анимацией.

Краткое содержание:

загрузка необходимых ассетов;
создание и настройка уровня;
создание анимации и Animation Controller.

Домашние задания

1 Углубление навыка работы с анимациями

Цель: Закрепить навыки работы с анимацией в Unity.

1. Возьмите файлы проекта из материалов к занятию или из GitHub-a
 2. Добавьте в Animator Controller анимацию смерти персонажа. Сделайте так, чтобы после удара атакуемый персонаж отыгрывал анимацию смерти
 3. Добавьте еще одно состояние для атаки кулаком (используйте анимацию из Z_attack_A).
-

2 Игровой цикл в Unity

Цели занятия:

познакомиться с особенностями скриптов и игрового цикла в Unity;
реализовать базовые игровые механики.

Краткое содержание:

основные методы в Unity скриптах;
игровой цикл;
coroutine.

Домашние задания

1 Сделать механику с корутиной

Цель: Навыки работы с корутинами

1. Придумайте механику, которая работает на корутине

- а.) Период выстрела оружия
- б.) Ожидание окончания хода
- в.) Перемещение или анимация объекта
- г.) Ваш вариант

2. Реализуйте в коде эту механику

3 Логика игрового уровня

Цели занятия:

продолжить знакомиться с программирование скриптов в Unity;
реализовать логику игрового уровня.

Краткое содержание:

условия победы и поражения;
несколько видов оружия;
бонусы/уникальность персонажей.

4 Unity UI и загрузка сцен

Цели занятия:

познакомиться с UI в Unity и реализовать меню.

Краткое содержание:

UI компоненты в Unity;
реализация переходов между сценами;
методом DontDestroyOnLoad;
реализация индикатора прогресса загрузки.

Домашние задания

1 Закрепление навыков работы с UI

Цель: В этой ДЗ закрепляем навыки создания пользовательских интерфейсов в Unity

1. Возьмите код из материалов к занятию
2. Сделайте не менее двух уровней в игре (один уровень - одна сцена).
3. Реализуйте выбор уровня в главном меню
4. В режиме игры добавьте в верхнем углу кнопку, открывающую на этом же экране меню с опциями "Продолжить", "Начать уровень заново", "Выйти в главное меню".
5. Добавьте всплывающие сообщения при выигрыше и при проигрыше с соответствующими сообщениями и кнопкой для возврата в главное меню.
- 6*. Реализуйте отображение здоровья персонажей с помощью Canvas в режиме World Space

5 Спецэффекты

Цели занятия:

познакомиться с настройкой и интеграций спецэффектов в проект.

Краткое содержание:

готовые ассеты эффектов;
система частиц в Unity, компонент ParticleSystem;
настройка эффектов камеры.

Цели занятия:

интегрировать звуки и музыку в игру;
создать меню настроек громкости.

Краткое содержание:

аудио ассеты в Unity;
компоненты AudioSource и AudioListener;
инструмент AudioMixer;
меню настройки громкости звука.

Домашние задания

1 Навыки работы со звуками

Цель: В этом ДЗ закрепляем навыки работы со звуками.

1. Возьмите код из материалов к занятию
2. Добавьте звуки для всех действий в игре: выстрел из пистолета, удар битой, удар кулаком, ранение персонажа, смерть персонажа, выигрыш, проигрыш.
3. Добавьте музыку в главное меню и игровые уровни (должно быть хотя-бы два разных музыкальных трека).
4. Добавьте настройку разных групп звуков.

1 Создание 2D проекта

Цели занятия:

познакомиться с разработкой 2D игр на Unity;
создать простой уровень.

Краткое содержание:

2D-проект в Unity;
создание и настройка спрайтов;
спрайтовая анимация;
Tilemap.

2 Введение в физику

Цели занятия:

познакомиться с инструментами 2D-физики в Unity и возможностями их использования в игре.

Краткое содержание:

компонент RigidBody2D;
коллайдеры;
физические материалы.

Домашние задания

1 Доработка управления персонажем и платформы

Цель: В этом домашнем задании закрепляем базовые навыки работы с 2D-физикой в Unity.

1. Скачайте проект из материалов к занятию
 2. Реализуйте прыжок от стены для персонажа. Если персонаж, находясь в воздухе, касается стены и нажимает на прыжок, он должен отпрыгнуть вверх и в сторону, противоположную стене, которой он касается.
 3. Сделайте так, чтобы при выпадении персонажа за пределы карты показывалось сообщение о проигрыше с кнопкой начала игры заново.
 4. Реализуйте платформу, которая может двигаться через несколько точек.
-

3 Продвинутая физика

Цели занятия:

познакомиться с продвинутыми элементами 2D-физики.

Краткое содержание:

компоненты joints;
компоненты area effectors.

4 2D анимация. DOTween, Spine и другие инструменты

Цели занятия:

познакомиться с дополнительными инструментами для 2D-анимации.

Краткое содержание:

скелетная 2D-анимация в Unity;
Spine анимация;
инструмент Anima2D;
библиотека DOTween.

5 Обработка ввода, New Input System

Цели занятия:

познакомиться с новой обработкой ввода в Unity и сравнить ее с классической системой ввода.

Краткое содержание:

классическая система ввода (класс Input);
новая система ввода (пакет Input System);
реализация корректного управления персонажем.

6 Игровой прогресс и его сохранение

Цели занятия:

реализовать сохранение и загрузку прогресса и настроек с использованием PlayerPrefs, JsonUtility и BinarySerializer.

Краткое содержание:

познакомимся с классом PlayerPrefs;
изучим использование JsonUtility для сериализации/десериализации компонентов;
посмотрим на инструмент BinarySerializer;
научимся организовывать архитектуру для сохранения и загрузки.

1 Знакомство с Terrain, ProBuilder

Цели занятия:

познакомиться с инструментами Terrain, ProBuilder;
создать уровень игры.

Краткое содержание:

инструмент Terrain для создания outdoor-уровней;
плагин ProBuilder;
создание уровня.

2 Организация структуры и архитектуры проекта

Цели занятия:

познакомиться с подходами к лучшей организации и проектированию проекта.

Краткое содержание:

познакомимся с возможными структурами проекта; познакомимся с архитектурой проектов на Unity.

Домашние задания

1 Добавить в вашу игру генератор уровня

Цель: Реализовать генератор уровней в проекте для закрепления навыков работы с генератором уровней

1. Скачайте проект из материалов к занятию
2. Создайте новый класс генератора уровней, объединяющий возможности блочного и клеточного генераторов. Половина блоков в сгенерированном уровне должны быть выбраны из заготовленных заранее, тогда как другая половина должна быть сгенерирована с помощью клеточного автомата.
3. Уделите внимание правильной стыковке блоков (не должно быть соединений вида "проход в стену")
4. Добавьте в генератор поддержку объектов в предзаготовленных блоках. Сделайте так, чтобы в некоторых предзаготовленных блоках были NPC и платформы.

3 Управление персонажем

Цели занятия:

создать контроллер управления персонажем и познакомиться со слоями анимации.

Краткое содержание:

контроллер персонажа; анимация, слои, блендинг.

4 **Враги, оружие.
Navigation
Mesh**

Цели занятия:

познакомиться с NavMesh компонентом;
реализовать врагов на уровне.

Краткое содержание:

Navigation Mesh, с использованием статических и динамических препятствий;
Navigation Agent для персонажей;

5 **Логика
игрового
уровня FPS**

Цели занятия:

реализовать логику игрового уровня.

Краткое содержание:

оружие и жизни;
условие победы и поражения на уровне.

Домашние задания

1 Добавление оружия типа projectile

Цель: Закрепление навыков работы с сетевым кодом

1) возьмите код проекта из материалов к занятию или с GitHub.

2) добавьте в игру тип оружия "ракетница": при выстреле создается ракета, которая летит по прямой и взрывается при столкновении с любым препятствием, нанося повреждение всем персонажам на небольшом удалении от места столкновения.

3) добавьте в игру тип оружия "миномет": при выстреле создается мина, которая ведет себя как физический объект - отскакивает от пола и от стен, но взрывается при столкновении с персонажем, а также - через заданное количество секунд, даже если не произошло столкновения с персонажем. При взрыве, также как и ракета, мина наносит повреждения всем персонажам вокруг эпицентра взрыва.

6 **Неигровые персонажи и Cinemachine**

Цели занятия:

познакомимся что такое неигровые персонажи в играх.
познакомимся с инструментом Cinemachine;

Краткое содержание:

неигровые персонажи;
добавление неигровых персонажей на уровень;
добавление и настройка камера с использованием Cinemachine.

7 **Локализация приложений**

Цели занятия:

узнать что же такое в действительности локализация и какие инструменты Unity предоставляет для этого.

Краткое содержание:

что такое локализация;
шрифты в TextMesh Pro;
right-to-left языки;
пакет локализации от Unity;
плагин локализации I2.

- | | | |
|---|---|--|
| 1 | Выбор темы и организация проектной работы | Цели занятия:

познакомиться с регламентом работы над проектом, сроками и выбором темы. |
| Краткое содержание: | | |
| правила работы над проектом;
требования к результату проекта и итоговой документации;
специфика проведения итоговой защиты. | | |
| <hr/> | | |
| 2 | Консультация по проектам и домашним заданиям | Цели занятия:

получить ответы на вопросы по проекту, ДЗ и по курсу. |
| Краткое содержание: | | |
| вопросы по улучшению и оптимизации работы над проектом. | | |
| <hr/> | | |
| 3 | Защита проектных работ | Цели занятия:

защитить проект и получить рекомендации экспертов. |
| Краткое содержание: | | |
| презентация проектов перед комиссией;
вопросы и комментарии по проектам; | | |
| Домашние задания | | |
| 1 Сдача итогового проекта | | |