**NFT-питомец, который умирает без ухода**

**Задача**: NFT-питомец, который умирает без ухода

**Описание:** NFT, у которого есть "здоровье". Владельцу нужно регулярно "кормить" его через транзакцию. Иначе NFT становится "мертвым".

Особенности:

Состояние NFT зависит от действий пользователя

Может эволюционировать или исчезнуть

Можно торговать питомцами

**🔹 Цель проекта**

Создать NFT-питомца, который требует регулярного ухода. Если пользователь не кормит питомца через смарт-контракт, он умирает. Проект объединяет игровую механику с NFT и показывает, как блокчейн можно использовать для создания привычек и вовлечения пользователей.

**🔹 Участники проекта**

### Инна – блокчейн и интеграция

**Обязанности:**

* Написать смарт-контракт на Solidity (NFT, здоровье, кормление, смерть).
* Протестировать контракт в Hardhat.
* Подключить контракт к сайту (через MetaMask).

### Алина – сайт и дизайн

**Обязанности:**

* Сделать веб-интерфейс на ASP.NET (страницы с питомцами).
* Подключить кнопки (например, «Кормить» или «Посмотреть»).
* Найти или нарисовать картинки питомцев (разные состояния).
* Сделать сайт понятным и красивым.

### 🔹 ****Технологии (технический стек)****

* **Solidity + Hardhat** — создание и тестирование смарт-контрактов.
* **ASP.NET** — создание сайта (интерфейса).
* **Web3.js / ethers.js** — подключение сайта к блокчейну.
* **MetaMask** — вход в систему и отправка транзакций.
* **IPFS** — хранение изображений питомцев.

**🔹 Основной функционал**

**1. NFT-питомец**

* NFT на основе стандарта **ERC-721**.
* Метаданные включают:
  + Имя питомца
  + Уровень здоровья (число от 0 до 100)
  + Дата последнего кормления (timestamp)
  + Состояние: active, evolved, dead
  + Этап развития (1, 2, 3)
  + URI на изображение питомца

**2. Механика здоровья**

* Питомец теряет **X** единиц здоровья каждые **N часов** (например, -10/24ч).
* Команда feedPet(uint256 tokenId) восстанавливает здоровье (например, +30).
* Если здоровье опускается до 0 → питомец становится **мертвым** (не может быть эволюционирован, продан, или накормлен).
* Если здоровье долгое время на 100% → питомец может **эволюционировать** (смена внешнего вида и повышение ценности).

**3. Эволюция**

* При достижении определенного уровня ухода (например, 7 дней здоровья 100/100):
  + NFT меняет статус на evolved.
  + Обновляется изображение (новый IPFS URI).
  + Можно предусмотреть несколько этапов: baby → adult → rare.

**4. Смерть и исчезновение**

* После смерти NFT нельзя кормить.
* Опционально: после **30 дней мертвого состояния** NFT можно **удалить** (burn).
* Возможность показать "кладбище питомцев" на фронте.

**5. Торговля**

* Игроки могут покупать новых питомцев.
* Можно **передавать** или **продавать** питомцев другим.

**🔹 Web-интерфейс (ASP.NET)**

**1. Главная страница**

* Краткое описание проекта
* Кнопка "Войти с MetaMask"
* Список своих питомцев

**2. Страница питомца**

* Отображение:
  + Имя, здоровье, стадия, статус
  + Изображение (в зависимости от стадии)
* Кнопки:
  + Feed — отправляет транзакцию на кормление
  + Sell / Transfer — продать или передать NFT
* Таймер до следующего снижения здоровья

**3. Магазин**

* Возможность купить нового питомца (mint).

**4. Кладбище**

* Галерея "умерших" питомцев с их именами и последним здоровьем.

### 🔹 ****Дополнительно (необязательно)****

* Напоминание, когда нужно покормить питомца.
* Баллы или достижения за хороший уход.
* Таблица лучших игроков.

**🔹 Смарт-контракт: основные методы**

function mintPet(string memory name) public; — создать питомца

function feedPet(uint256 tokenId) public; — покормить

function getPetStatus(uint256 tokenId) public view returns (HealthStatus memory); — посмотреть статус

function evolvePet(uint256 tokenId) public; — эволюционировать

function burnDeadPet(uint256 tokenId) public; — удалить мёртвого питомца

**🔹 Состояния питомца**

| **Статус** | **Условия** | **Возможные действия** |
| --- | --- | --- |
| Active | Здоровье > 0 | Кормить, эволюционировать |
| Evolved | Выполнены условия эволюции | Торговать |
| Dead | Здоровье = 0 | Только просмотр / burn |

**🔹 Безопасность**

* Ограничения на частоту кормлений
* Только владелец может кормить или удалять питомца.

**🔹 Этапы разработки**

| **Этап** | **Что делаем** |
| --- | --- |
| 1. Подготовка | Придумываем дизайн, список функций |
| 2. Контракт | Пишем NFT-контракт и логику кормления/смерти |
| 3. Интерфейс | Делаем сайт на ASP.NET |
| 4. Интеграция | Подключаем сайт к MetaMask и контракту |
| 5. IPFS | Загружаем картинки питомцев |
| 6. Торговля | Делаем возможность продавать и передавать |
| 7. Тестирование | Проверяем, чтобы всё работало |
| 8. Публикация | Показываем готовый проект |