1. **Первый урок: Hello world**

**Срок написания**: 1 день - смарт-контракт и тестирование на gtest готовы, нужно сделать раздел с idea и gclient (хотя возможно gclient можно пока не показывать).

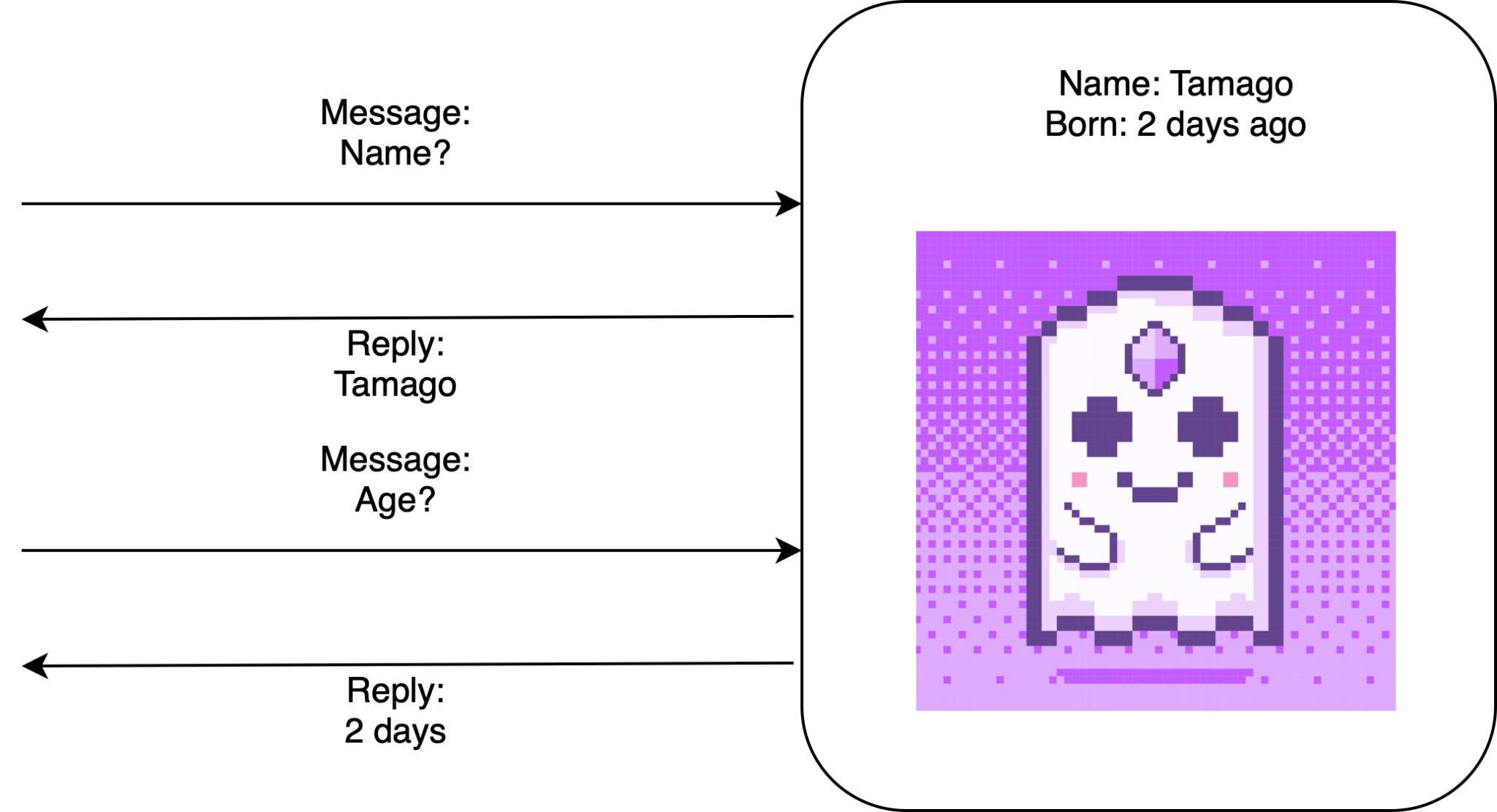
**Цель первого урока**:

* показать минимальную структуру программы на Gear: инициализация программы, точка входа программы, библиотека gstd;
* Показать, что программа - это отдельный actor, который может отвечать на сообщения, а также посылать сообщения;
* Показать обработку нескольких сообщений;
* Введение в gtest (можно добавить работу с idea и gclient).

**Самостоятельная работа**:

Начинаем писать игру тамагочи:

* Создать структуру MyTamagotchi, которая будет хранить имя и дату рождения тамагочи;
* При инициализации установить его имя и дату рождения и послать ответ об успешной инициализации;
* Программа тамагочи должна принимать следующие сообщения:
* Name - программа отвечает, как зовут тамагочи;
* Age - программа отвечает про возраст тамагочи.



1. **Второй урок: Escrow**

**Цель второго урока:**

* Управление логикой программы в зависимости от состояния и времени;
* Функции из gstd библиотеки;
* Сообщения с прикрепленными value, проверка баланса программы;
* Тестирование программы с помощью gtest;
* Чтение стейта программы.

**Самостоятельная работа:**  
Игра тамагочи:

* Расширение стейта: добавить в структуру следующий поля:
* Хозяин тамагочи (Например, аккаунт, инициализирующий контракт);
* Stages - на каком этапе взросления находится тамагочи (Например: Baby, Child, Teenager, Adult ..);
* Состояния - сколько у тамагочи энергии (energy), насколько он сыт (fed) (так же можно добавить ,насколько он отдохнувший, развлеченный, потренировавшийся итд);

Эти значение нужно установить ненулевыми при инициализации тамагочи.

* Также при инициализации определить следующие константы:

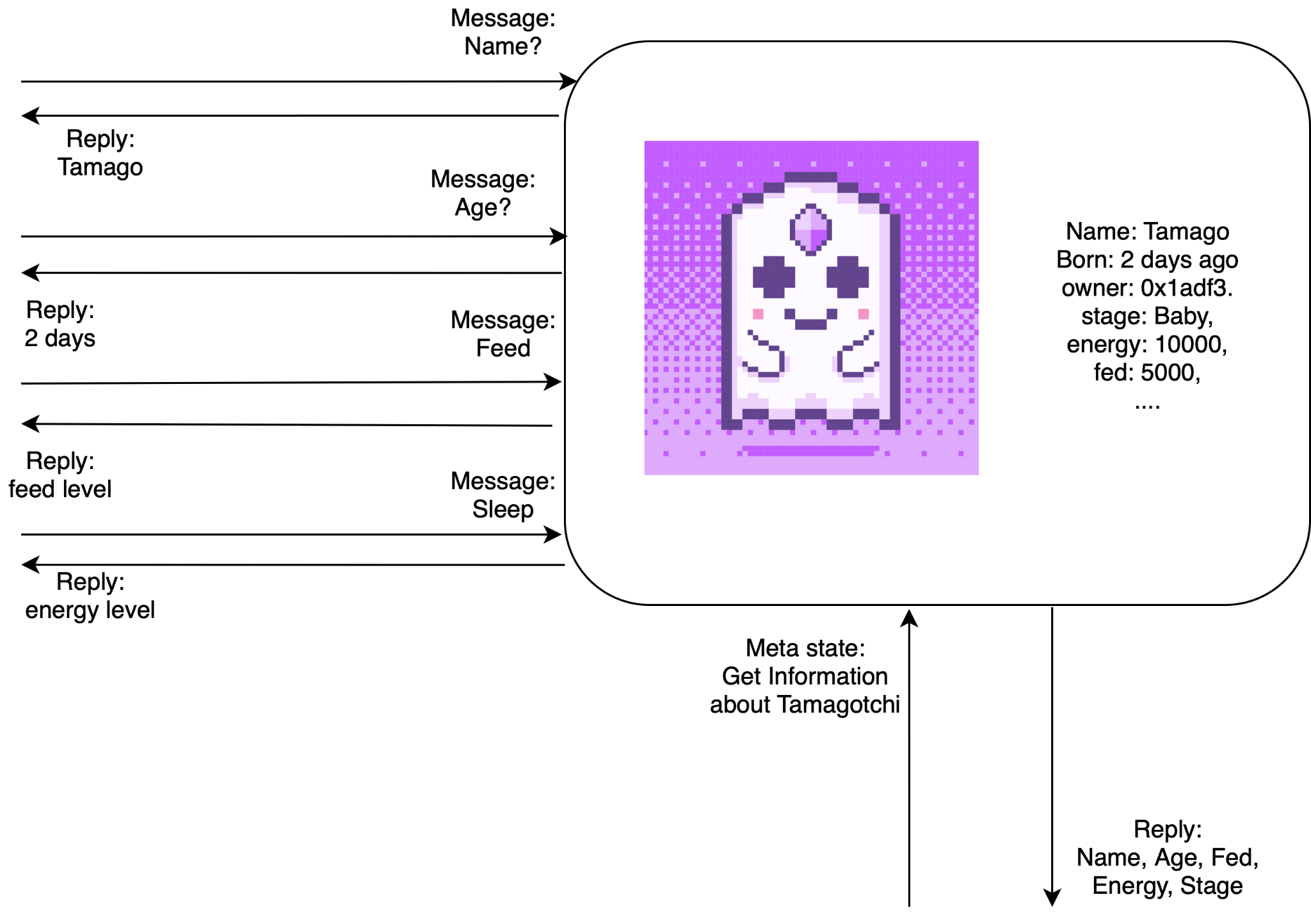
**hungerPerBlock** - насколько тамагочи становится голодным за блок;

**energyPerBlock** - сколько тамагочи теряет энергии за блок;

**energyPerSleep** - сколько тамагочи получает энергии за сон;

**FillWithFeed** - насколько тамагочи становится сытым за одно кормлениe;

* Тамагочи так же теперь должен принимать сообщения: Sleep и Feed (Здесь должна быть реализована логика подсчета, насколько тамагочи проголодался/устал до сообщений и насколько он сыт/отдохнул после);
* Добавить чтение состояния тамагочи в текущий момент (имя, возраст, stage, насколько он сыт, сколько у него энергии).



**Что нужно добавить в уроке**: чтение состояние программы и тестирование состояния программы (добавится, когда будет готова новая метадата).

1. **Третий урок: Factory Escrow (move to addition resources)**

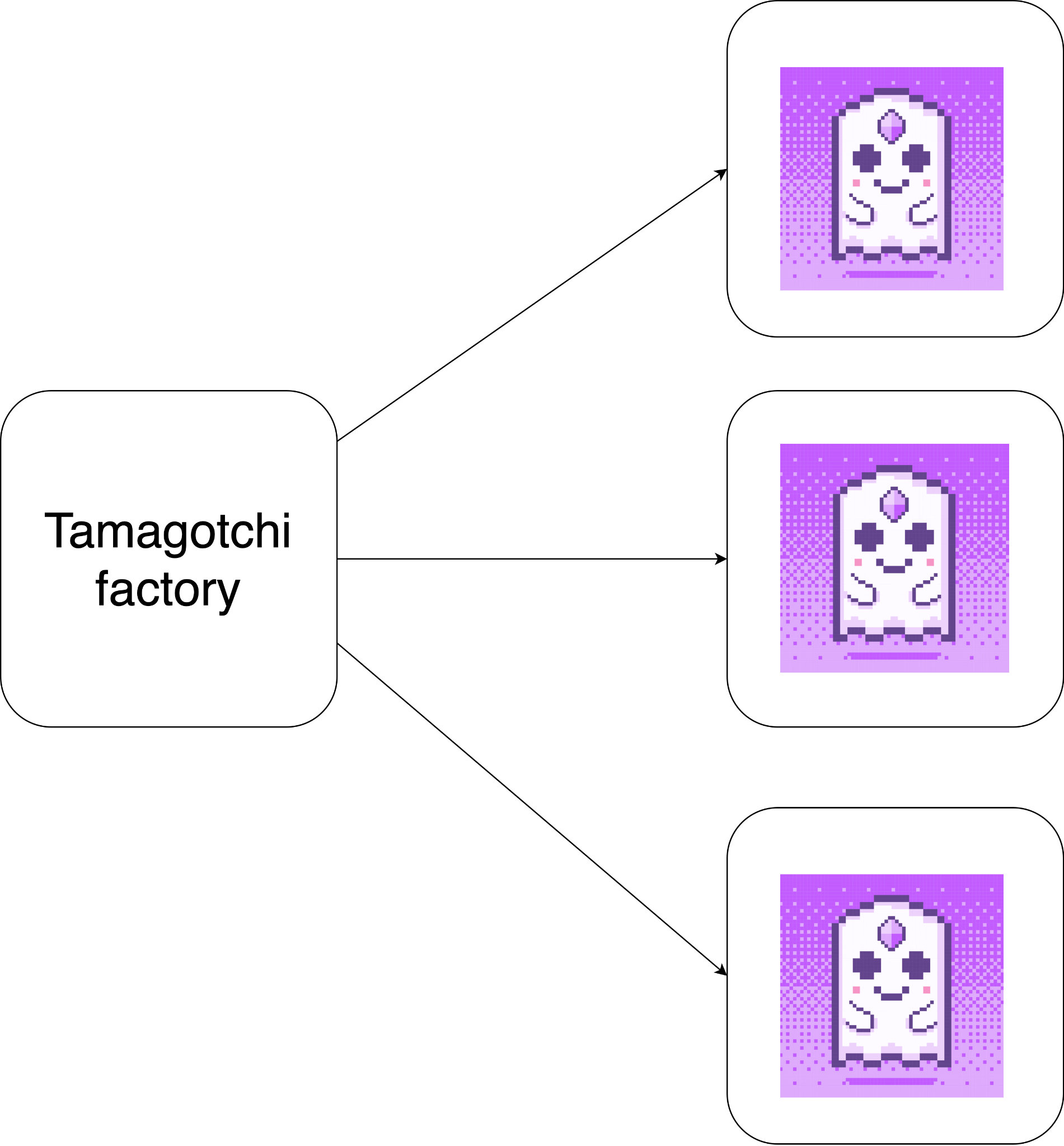
**Срок написания**: меньше дня - смарт-контракт и тестирование на gtest готовы, нужно прочитать, возможно нужно будет подправить текст по замечаниям

**Цель урока:**

* создание программы из другой программы;
* Тестирование этой логики в gtest и в idea.

**Самостоятельная работа:**

* Написать программу, которая будет создавать тамагочи;
* Программа должна сохранять адреса созданных тамагочи, а также создателей тамагочи



1. **Четвертый урок: Fungible token/Non-fungible token**

**Срок написания**: до 2х дней - смарт-контракт sharded fungible token готов, нужно будет очень понятно и доступно здесь объяснить по состояния программ и асинхронные сообщения. Так же нужно будет объяснить использование библиотеки gear-lib для написания NFT (и возможно в gear-lib добавится sharded fungible token).

**Цель урока**:

* Объяснение fungible-token на Gear (новый sharded fungible token);
* Non-fungible token на Gear.
* Работа с асинхронными сообщениями.

**Самостоятельная работа** - добавить в контракт Tamagotchi Factory функционал NFT:

* Создание тамагочи (уже есть);
* Передача тамагочи от одного пользователя к другому;
* Разрешение какому-либо аккаунту распоряжаться тамагочи.

А также добавить в контракт тамагочи метаданные (поле для ссылки на картинку). Изначально картинка тамагочи может быть примерно у всех одинаковая, но со временем ее можно будет менять, покупая разные атрибуты (следующий урок).

1. **5 урок: Tamagotchi shop**

**Срок написания**: до 2х дней - смарт-контракта нет, нужно написать урок полностью.

**Цель урока:**

* Написать контракт, в котором тамагочи смогут покупать себе какие-нибудь атрибуты: элементы одежды, оружие, зелье для сражений (сражения появятся в следующих уроках).
* Показать работу с асинхронными сообщениями на примере взаимодействия с fungible token.

**Самостоятельная работа:**

* Добавить в контракт тамагочи функционал покупки, и соответственно обновление внешнего вида и характеристик тамагочи в зависимости от покупок.

1. **6 урок: Аукцион**

**Срок написания**: до 2х дней - смарт-контракта нет (за основу смарт-контракта возьмется наш пример с gear-DApps), нужно написать текст урока.

Цель урока:

* Написать контракт, в котором пользователи смогут выставлять тамагочи на аукцион, а также на продажу;
* Показать работу с отложенными сообщениями на примере аукциона;

**Самостоятельная работа:**

* Добавить в тамагочи возможность отправлять пользователю сообщения себя покормить, обновить и так далее через отложенные сообщения.

1. **7 урок: Tamagotchi Battle**

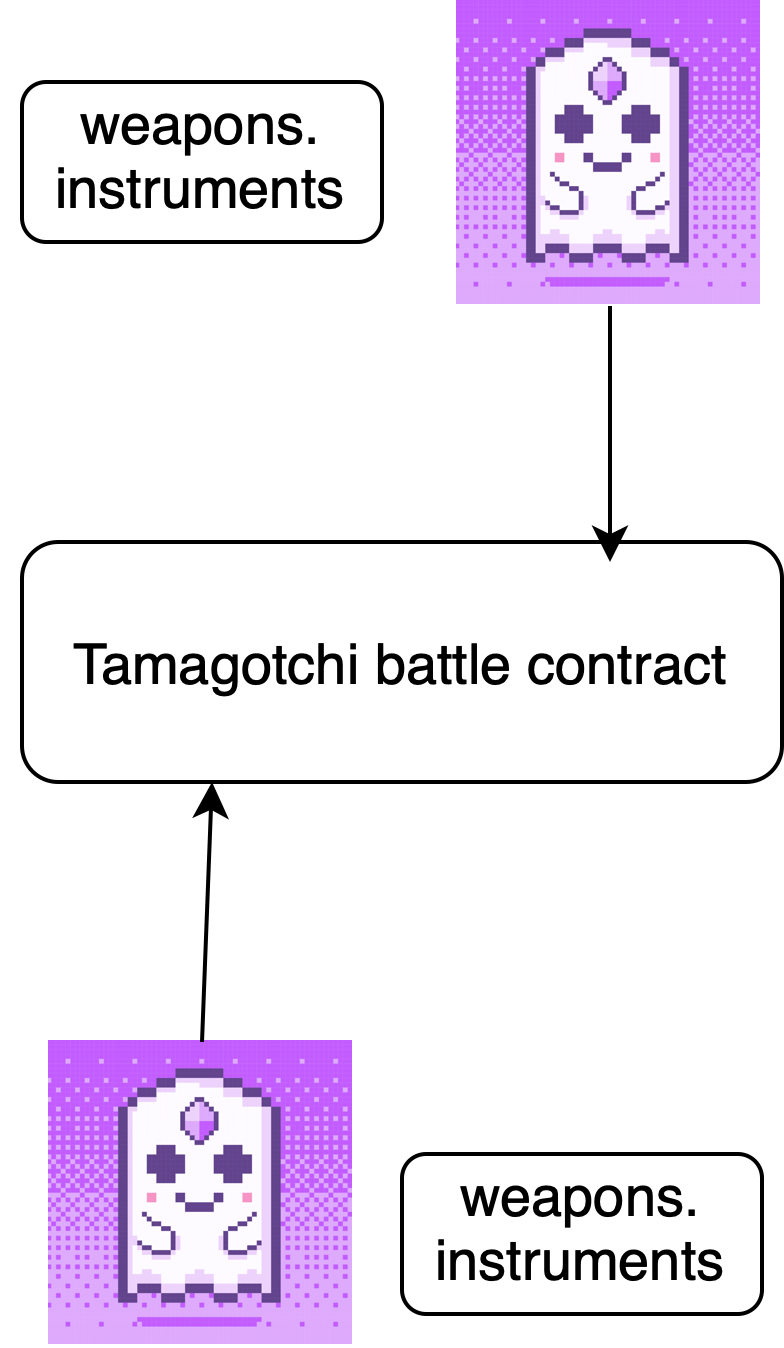
**Срок написания**: до 3х дней - это минимально начальная игра, нужно тут согласовать сложность механики. Если не слишком усложнять, до написание урока полностью - до двух дней, однако в этом уроке очень бы хотелось добавить графику, хотя бы 2D.

**Цель урока:**

* Добавить в функционал тамагочи возможность играть (сражаться) с другим тамагочи (аналог worms/wormix)

**Самостоятельная работа:** усложнения игры

* У тамагочи есть арсенал оружия и инструментов, который он можем купить в контракте магазина (5 урок). Самые простые оружия могут быть с самого начала (или при достижении определенного возраста). В дальнейшем оружия можно улучшать;



.