1. 第一课：你好世界

截止日期：1 天 - 智能合约和 gtest 测试已经准备就绪，你需要制作一个包含 idea 和 gclient 的部分（尽管可能还不能显示 gclient）。

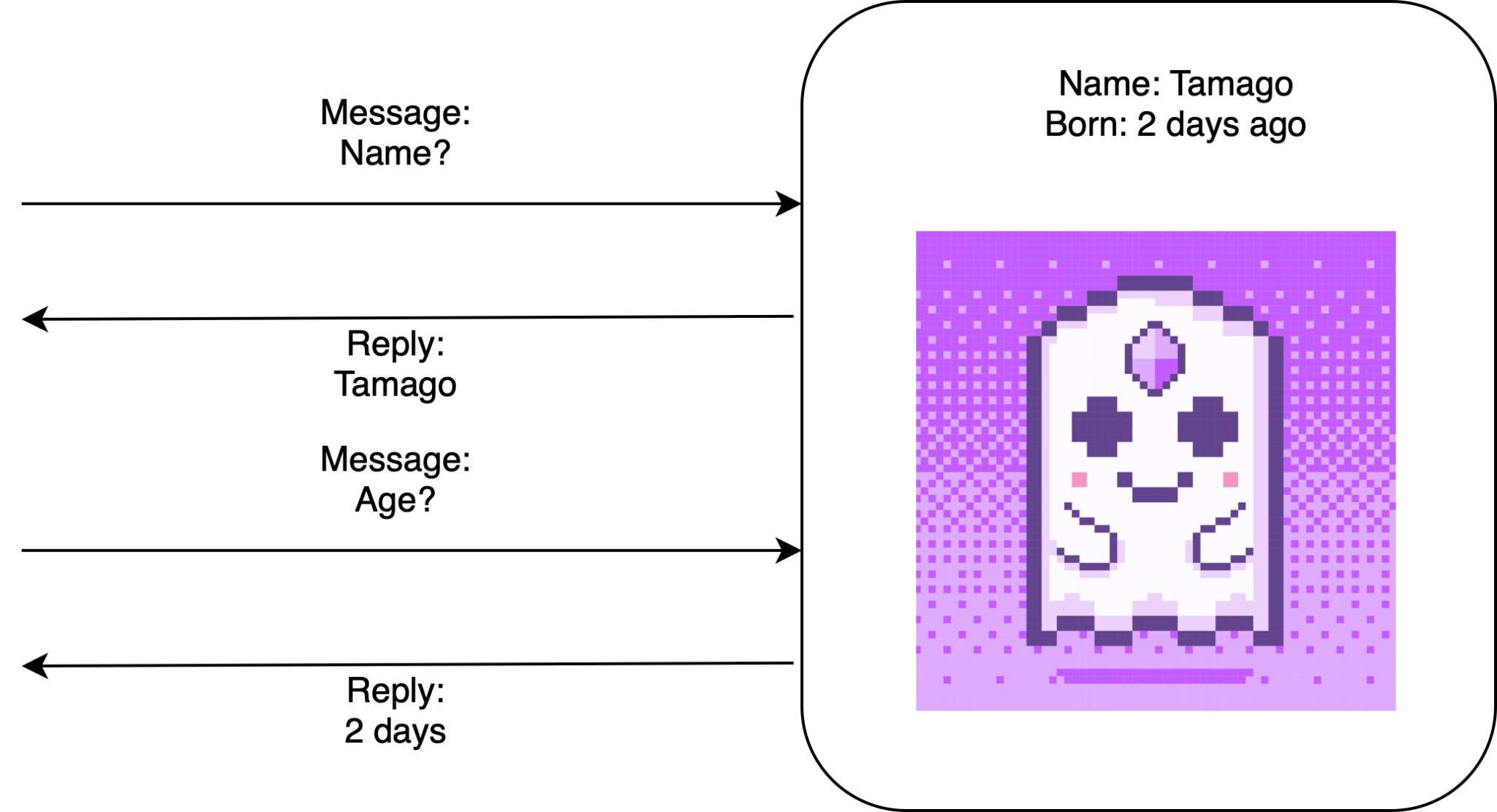
第一节课的目的：

* 显示 Gear 程序的最小结构：程序初始化、程序入口点、gstd 库；
* 表明该程序是一个独立的参与者，既可以响应消息也可以发送消息；
* 显示多条消息的处理；
* gtest 简介（可以配合 idea 和 gclient 加 work）。

独立编码工作：

让我们开始写一个 Tamagotchi 游戏：

* 创建一个 MyTamagotchi 结构，用于存储 Tamagotchi 的名称和出生日期；
* 初始化时，设置其名称和出生日期，并发送初始化成功的响应；
* Tamagotchi 程序应该接受以下消息：
* 名称 - 程序回答电子宠物的名称；
* 年龄 - 该程序会回答电子宠物的年龄。



1. 第 2 课：托管

第二课目的：

* 根据状态和时间进行程序逻辑控制；
* gstd 库中的函数；
* 有附加值的消息，程序余额检查；
* 用 gtest 测试程序；
* 读取程序状态。

自学：电子鸡游戏：

* 状态扩展：在结构中添加以下字段：
* Tamagotchi master（例如初始化合约的账户）；
* 阶段 - Tamagotchi 处于哪个成长阶段（例如：婴儿、儿童、青少年、成人..）；
* 状态——Tamagotchi 有多少能量（energy），他有多饱（fed）（你也可以加上他是如何休息、娱乐、训练等）；

初始化 Tamagotchi 时，此值必须设置为非零。

* 此外，在初始化期间，定义以下常量：

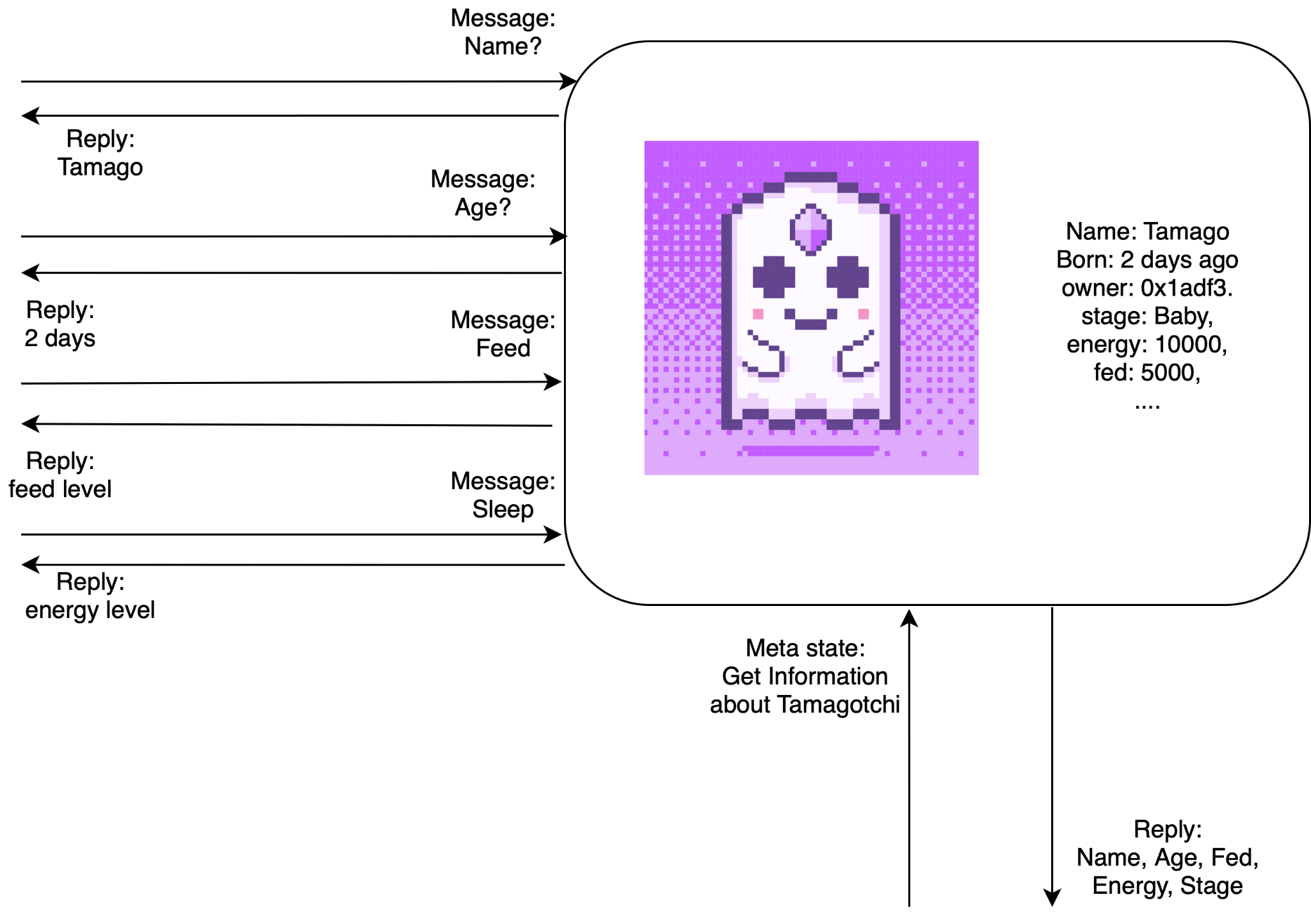
hungerPerBlock - Tamagotchi 每个方块的饥饿程度。

energyPerBlock - Tamagotchi 每块损失多少能量；

energyPerSleep - 电子宠物每次睡眠获得多少能量；

FillWithFeed - Tamagotchi 在一次喂食中的饱满程度；

* Tamagotchi 现在也应该接受消息：Sleep 和 Feed（这里，应该实现计算消息前饥饿/疲劳程度和消息后饱腹/休息程度的逻辑）；
* 添加当前电子宠物状态的读数（名称、年龄、阶段、他有多饱、他有多少能量）。



课程中需要添加的内容：读取程序的状态和测试程序的状态（将在新的元数据准备就绪时添加）。

1. 第 3 课：工厂托管（移至其他资源）

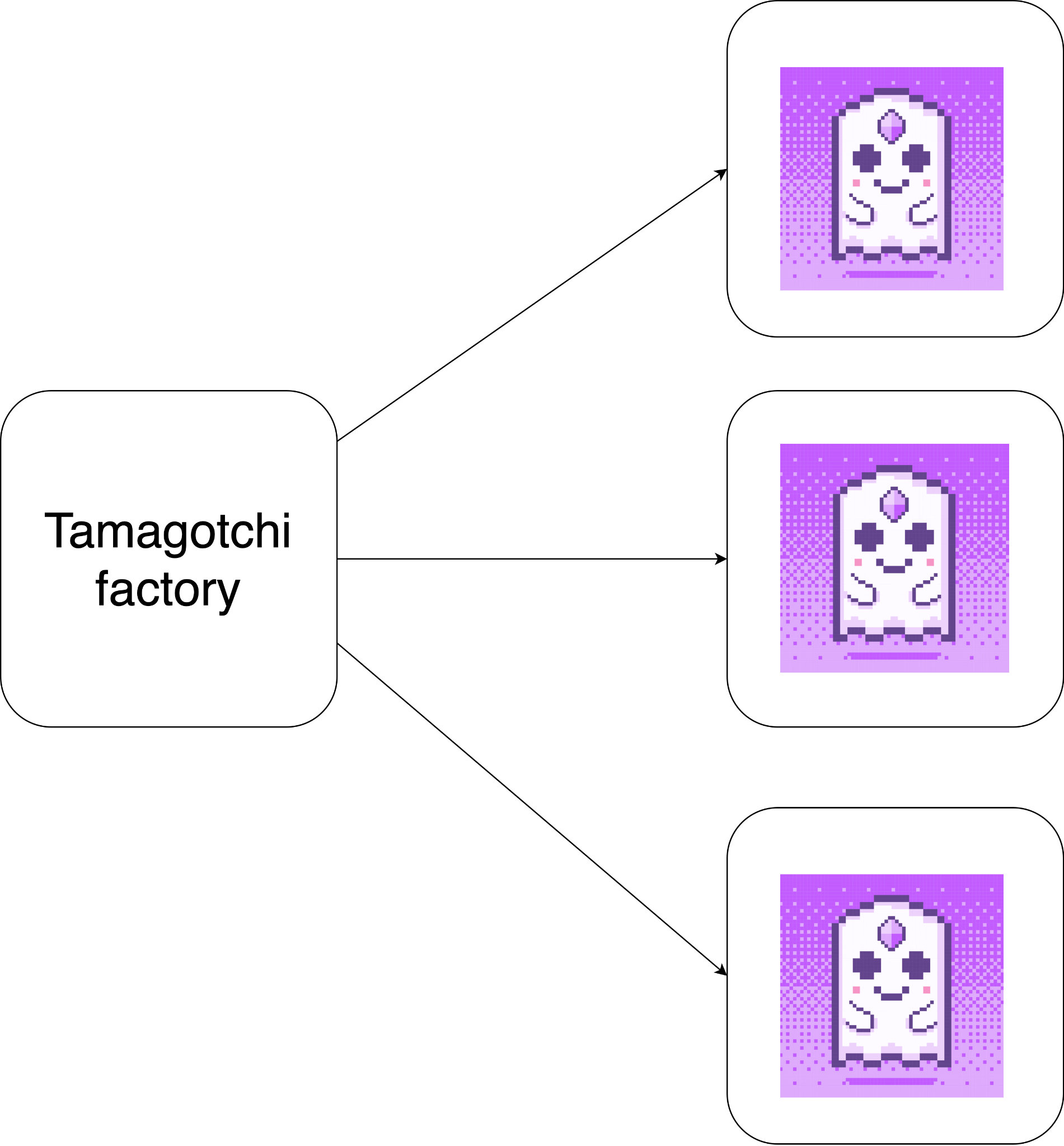
截止日期：不到一天 - 智能合约和 gtest 测试已经准备就绪，需要阅读，可能需要根据评论修改文本

课程目的：

* 从另一个程序创建一个程序；
* 在 gtest 和 idea 中测试这个逻辑。

独立编码工作：

* 编写一个程序来制作电子鸡；
* 该程序必须保存创建的电子鸡的地址，以及电子鸡的创建者



1. 第四课：Fungible token/Non-fungible token

截止日期：最多 2 天 -分片可替代令牌智能合约已准备就绪，有必要以非常清晰易懂的方式解释程序状态和异步消息。你还需要解释如何使用  
gear-lib 库来编写 NFT（并且可能会向 gear-lib  
添加一个分片的可替代令牌）。

上课目的：

* Gear 上的 fungible-token 说明（新的分片可替代令牌）；
* Non-fungible token - Gear。
* 使用异步消息。

自己动手——将 NFT 功能添加到 Tamagotchi Factory 合约中：

* Tamagotchi 的创作（已经存在）；
* 将 Tamagotchi 从一个用户转移到另一个用户；
* 允许任何帐户管理 Tamagotchi。

并且还将元数据添加到 Tamagotchi 合约（用于链接到图像的字段）。最初，Tamagotchi 图片对每个人来说可能大致相同，但随着时间的推移，它可以通过购买不同的属性来改变（下一课）。

1. 第 5 课：电子鸡店  
   截止日期：最多 2 天 - 没有智能合约，你需要完整地编写课程。

上课目的：

* 写一份合约，其中 Tamagotchis 将能够为自己购买一些属性：衣服元素、武器、战斗药水（战斗将在下一课中出现）。
* 使用与可替代令牌交互的示例展示异步消息的工作。

独立编码工作：

* 将购买功能添加到 Tamagotchi 合约中，并根据购买情况相应地更新 Tamagotchi 的外观和特征。

1. 第 6 课：拍卖

截止日期：最多 2 天 - 没有智能合约（我们以 gear-DApps 为例将作为智能合约的基础），你需要写课文。

上课目的：

* 编写一份合约，用户可以在合约中将电子宠物用于拍卖和出售；
* 在拍卖示例中显示带有延迟消息的作品；

独立编码工作：

* 为 Tamagotchi 添加向用户发送消息以通过延迟消息喂养自己、更新等的能力。

1. 第 7 课：宠物大战

截止日期：最多 3 天 -这是最短的初始游戏，你需要在这里就机制的复杂性达成一致。如果不是太复杂，在完全写完课程之前

* 最多两天，但在这节课中我非常想添加图形，至少是 2 天。

上课目的：

* 将与另一个 Tamagotchi 玩耍（战斗）的能力添加到 Tamagotchi 功能（类似于蠕虫/wormix）

DIY：游戏并发症

* Tamagotchi 拥有可以在商店合约中购买的武器和工具库（第 5 课）。最简单的武器可以从一开始（或达到一定年龄），未来可以改进武器

