

## Тестовое задание для кандидата на позицию Ведущего Unity-разработчика

**Цель:** продемонстрировать навыки построения архитектуры и работы с мультиплеером в рамках ограниченного прототипа онлайн настольной игры.

### **Задача:**

Сделать прототип мультиплеерной “крестики-нолики” (Tic-Tac-Toe) на 2 игрока с использованием одного из следующих сетевых стеков:

Photon Fusion / Photon PUN2

Unity Netcode for GameObjects + Relay + Lobby

### **Минимальные требования:**

1. Мультиплеерная сессия

Игрок А создает комнату / лобби

Игрок В подключается через код комнаты или автоматический матчмейкинг

После подключения — начинается матч 1 на 1

### **2. Игровой процесс**

Стандартные правила игры “Крестики-нолики”

Синхронизация ходов между игроками

Победитель определяется и отображается обоим игрокам

### **3. Архитектура**

Использование Zenject или ViContainer (на выбор)

Использование UniRx или R3 для реактивного взаимодействия (на выбор)

### **4. UI**

Простое UI с отображением:

Кто сейчас ходит

Результат матча

Кнопка "Начать заново" (локальный рестарт в рамках текущей сессии)

### **5. Бонус (по желанию)**

Использование Addressables для подгрузки сцены или игровых префабов

Автоматический рестарт игры после завершения (по согласию обоих игроков)

Логику отделить от визуальной части — минимальный MVVM подход

### **Оценка:**

- Мы будем оценивать:
- Чистоту и модульность кода
- Структуру проекта и работу с зависимостями
- Использование реактивного подхода
- Правильную реализацию сетевой логики
- Умение разделять зоны ответственности
- Простоту и логичность взаимодействия компонентов

### **Требования к сдаче:**

- Git-репозиторий (GitHub или архив)
- Инструкции по запуску (README)
- Краткое текстовое описание архитектурных решений (в README или отдельным файлом)
- Версия Unity 2022.3.36f1

### **Ориентир по времени выполнения:**

~4–6 часов на реализацию. Мы понимаем, что кандидаты заняты, поэтому ждем чистый MVP, а не polished UI или production-уровень фичей.