

Анализ задания

1. Недостатки и недоработки.

Очень много var, порой в самых очевидных местах, как в циклах for. Очень часто копируются такие системы как “Food Transfer”, “Food Visualizer” и прочее.

FoodPlace содержит переменные, отвечающие за время готовки и сгорания блюда. Получается, что время приготовления зависит не от блюда, а от кухонного прибора (сковородки), причем у каждой сковороды можно выставить своё время приготовления/подгорания.

Очень много взаимодействия между объектами идет через их имена, т.е. string переменные, они очень часто сравниваются. Опечатки в коде и самом редакторе Unity приводят к ошибкам. Возможно лучше кешировать имена блюд, чтобы снизить вероятность опечаток в коде.

Заметил ошибку в задании. В разделе где описывается механика приготовления хот-дога. “После тапа по свободной дощечке или хлебнице, появляется булочка для Хот-догов” и “Приготовление бургера идентично приготовлению хот-дога” - в прототипе, как и в самой игре (играл на iOS) приготовление этих блюд начинается только с тапа по хлебнице. Тап по пустой дощечке не приводит к выкладываю булочек нее.

2. Сомнительные решения для крупных проектов

Вместо контроллеров можно использовать ScriptableObjects, чтобы не создавать лишних GameObject и вообще не плодить MonoBehaviour классы. Помимо этого в проекте применяется Singleton паттерн, который последний пару лет ругают все, кому не лень из-за не явной связи с другими объектами и сложности тестирования.

Можно использовать структуры вместо некоторых классов.

Каждый раз, когда нужно будет отдать заказ, происходит сортировка посетителей по времени ожидания”. При большом числе одновременных посетителей это может стать проблемой. Следует добавлять посетителей через Queue чтобы иметь доступ к самому первому посетителю, причем CustomerPlaces можно оставить списком, либо вообще упростить до Array.

В тестовом задании не применяется “пул объектов”, что тоже может существенно производительность при росте проекта.

3. Как увеличить производительность на слабых устройства

Упаковать спрайты в атлас, чтобы снизить нагрузку при рендере (уменьшить Batches). Сейчас каждый спрайт отрисовывается отдельно.

4. Где можно изменить числовые параметры. Время ожидания посетителей, время готовки/сгорания блюд, количество посетителей?

CustomresController позволяет настроить числовые параметры для постетителей. FoodPlace - время готовки/сгорания.

5. Что нужно сделать, чтобы реализовать механику луковых колец?

В качестве аналога можно рассмотреть механику приготовления пончиков из первых уровней игры. В соответствии с заданием нужно будет внести некоторые коррективы, а именно: должен отсутствовать

отдельный FoodPlacer (аналог BurgerBunPlacer), т.к. тап необходимо делать именно по фритюрнице, блюдо должно автоматически перемещаться на стол.

- Подготовить спрайты фритюрницы и блюда (сырое, готовое, сгоревшее и сервированное), а также его префаб.
- Внести блюдо с XML таблицу заказов **orders**
- Создать фритюрницу с коллайдером. Повесить на нее FoodPlacer, FoodPlace и FoodTransfer. В FoodPlace поставить таймер на 5 секунд. Связать окончание таймера с функцией TryTransferFood в FoodTransfer. Можно сделать различными методами, например через события. Для оригинальных блюд эта функция вызывается по тапу на сковороду.
- Переопределить TryPlaceFood таким образом, чтобы выкладывались одновременно 3 блюда на все 3 FoodPlace (доски или тарелочки для отдельных порций луковых колец).