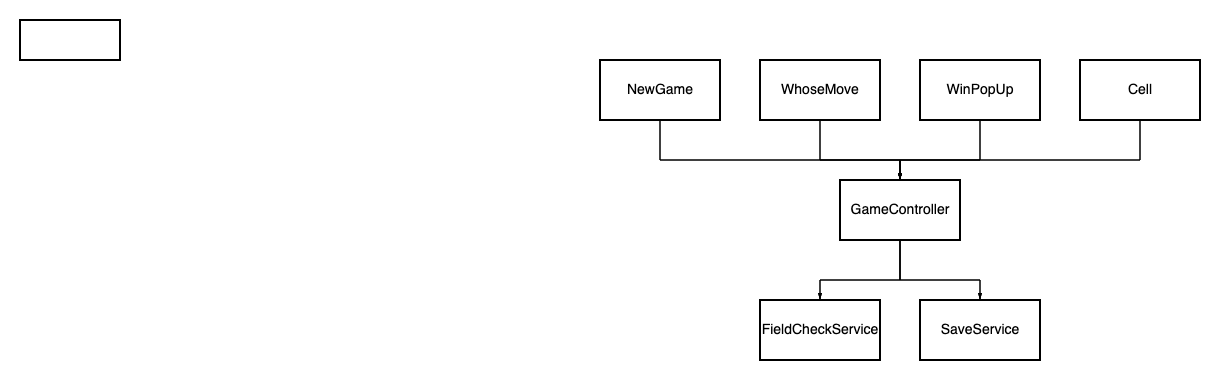
**Техническая Записка к Тестовому Заданию**

**TicTacToe**

**Архитектура игры**

Архитектура игры представлена на схеме



GameController – “Центральный” класс который хранит нынешнее состояние игры и принимает решение о дальнейшем развитии игры путем создания событий: Продолжать игру, объявить победителя или объявить ничью.

FieldCheckService – класс, ответственный за анализ игровой ситуации и поиска выигрышных комбинаций.

SaveService – класс, ответственный за сохранение, загрузку и рессет данных.

Cell – класс клекти, который отслеживает нажатие на себя и сообщает об этом GameController.

NewGame(UI) – класс, ответственный за передачу сигнала в GameController о необходимости перезапуска игры.

WinPopUp(UI) – класс, ответственный за вывод результата игры путем “Наблюдения” за GameController.

WhoseMove(UI) – класс, ответственный за вывод информации о том, чей сейчас ход, путем “Наблюдения” за GameController.

**Загрузка игры**

Первоначально на сцене имеется только MainCamera, Canvas, EventSystem, SceneContext и Инстайлеры.

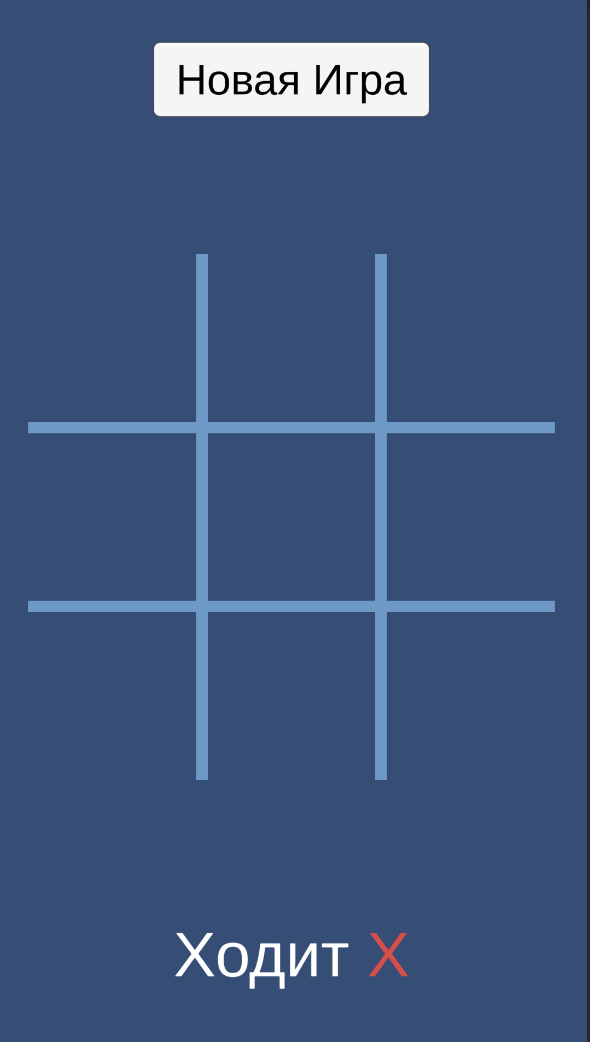
SceneContext по очереди подгружает инстайлеры:

1. ConfigInstaller – инжектит конфиги;
2. ServiceInstaller – инжектит сервисы;
3. GamePlayInstaller – инжектит игровый элементы;
4. UIInstaller – инжектит пользовательский интерфейс;

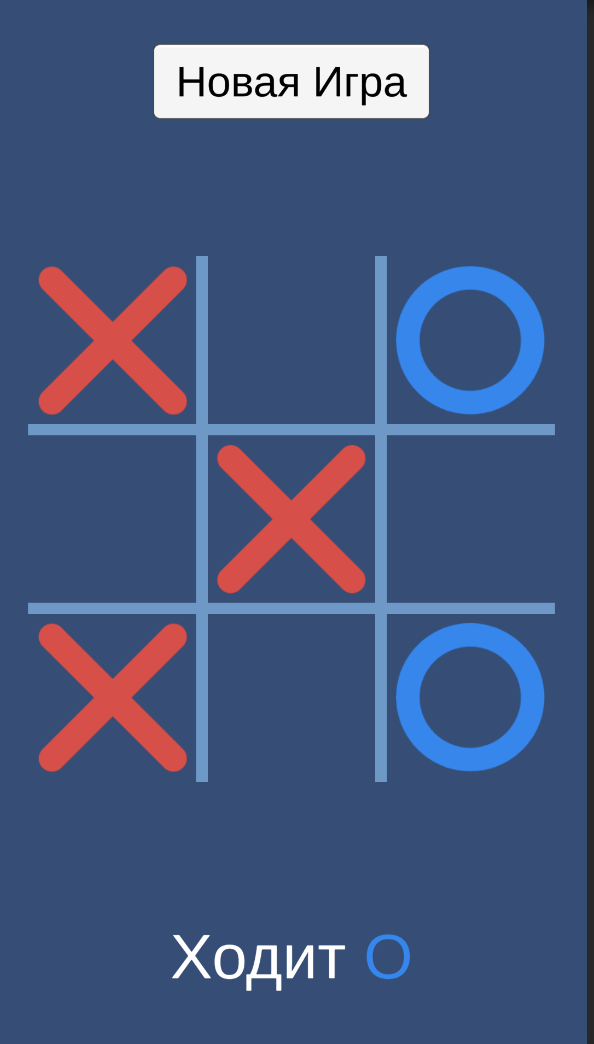
Все инстайлеры находятся в SceneContext, кроме UIInstaller. Он прикреплен к Canvas для быстрого и легкого назначения родителя интерфейсу.

После загрузки игры классы в Awake подписываются на необходимые события, а GameController в старте подгружает сохраненные данные. После чего начинается сам геймплей.

Стартовый экран (при отсутствии сохранений):



Процесс игры:



Возможные результаты конца игры:

