

Требования к билду:

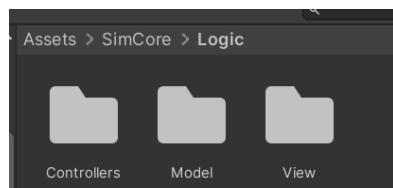
1. Name_version_vX.Y пример "superproject_build_v10.0_2023_01_01"
 - a. Где X - версия релиза
 - b. Где Y - незначительные изменения внутри релиза
 - c. Где v - версия продкута
 - d. Где дата - дата релиза
2. Только латиница
3. Ссылка на гит
4. Win desktop версия
5. Папка ProjectInfo в корневой директории, которая:
 - a. Содержит в свободной пронумерованной форме ошибки билда (файл BUGS.TXT)
 - b. Используемые крутые фичи (FEATURES.TXT)

Требования к репозиторию Git:

1. Все коммиты внутри репозитория проекта в аккаунте **OneRobot в GitHub**
2. Если разработчик работал в своем репозитории нужно сделать **трансфер**
3. Ссылка
4. Обязательно включить:
 - a. Билд
 - b. Проект
 - c. Readme (базовое описание)
 - d. BUGS.TXT
 - e. FEATURES.TXT
 - f. Gitignore файл, который позволяет коммитить билды
 - g. Поддержка LFS для GitHub (поддержка больших файлов больше 100 мб)

Требования к разработке на UNITY :

1. Выдерживать общий паттерн MVC (или любую из его вариаций MVP, MVVC и т.п.).

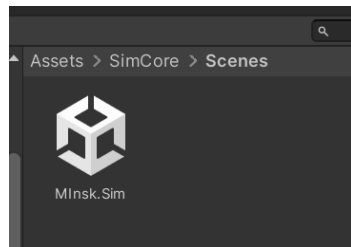


Примечание : мы за правильный код только когда это нужно.

Концепция чистый код - ради кода не для нас.

Зачем это нужно - для модульности и взаимозаменяемости и повторного использования, т.е. подобный подход используется только для потенциально важных фич. Простая view логика не так важна.

2. Сцены в директории Scenes (в корневой директории SimCore / GameCore)
 - a. ***НАЗВАНИЕ_ПРОЕКТА***_MAIN.Game - для игры
 - b. ***НАЗВАНИЕ_ПРОЕКТА***_MAIN.Sim - для симуляции



3. Требования к скриптам:
 - a. Едит в одном файле
 - b. Все скрипты внутри корневой директории LOGIC
 - c. Каждый класс - **в отдельном файле**
 - d. Namespace для **каждого класса**
 - e. Разбивать по папкам по функциональной части (в одной папке не должно быть больше 5 скриптов)
4. Требования к кассетам:
 - a. Путь на латинице
 - b. Название ассетов на латинице без пробелов
 - c. Таблица ключ/путь
5. Требование к префабам
 - a. Отдельная папка в корневой директории SimCore / GameCore
 - i. 1 префа = 1 сущность = 1 папка
 - ii. Вариативные префабы в дочерней папке основного префаба
 - iii. Из скриптов только MonoBeh без сложной доменной логики
6. Требование к сцене в Unity:
 - a. Разделение объектов внутри сцены:
 - i. ENV STATIC (окружение)
 - ii. LOGIC
 - iii. ENV DYNAMIC
 - iv. UI
7. Требование к репозиторию
 - a. Ветки
 - i. MAIN
 - ii. DEV
 - iii. FEATURE
 - b. Комит каждый день (если есть хоть 1 изменение)