

Архитектура проекта

Введение

Данный документ описывает архитектуру прототипа 3D аркадной игры в стиле Pacman, разработанного на движке Unity для платформы Android. Основной целью архитектуры является создание чистого, расширяемого и тестируемого кода через строгое разделение ответственностей между компонентами.

Используемые технологии и паттерны

- **Язык:** C#
- **Паттерн:** Адаптированный MVVM (Model-View-ViewModel) + Reactive Programming
- **Фреймворки:**
 - **UniRx** (Reactive Extensions for Unity)
 - **Zenject/Extenject** (Dependency Injection)
- **Навигация:** Unity NavMesh

Ключевые компоненты архитектуры

Модель (Model)

Чистые классы данных с состоянием игры:

- `GameData` - содержит реактивное свойство State (Play, Pause, Win, Lose)
- `PlayerData` - содержит реактивное свойство Score, логику перезагрузки

Представление (View)

Визуальные компоненты и UI:

- `Window`, `EndGameWindow`, `PauseWindow`, `ControlView` - UI
- `FixedJoystick` - управление
- `TextMeshProUGUI` - отображение счета

View-Model

Посредники между Моделями и Представлением:

- `ControlsViewModel` - логика победы, управление UI, преобразование input, обновление счета
- `EndGameWindowViewModel` - логика окна окончания игры
- `PauseWindowViewModel` - логика окна паузы игры

Сервисы (Services)

Абстракции для специфического функционала:

- `IPositionService` - интерфейс положения в мире
- `PlayerPositionService` - реализация для игрока
- `IMaxScoreService` - интерфейс для сохранения и загрузки максимального количества очков игрока
- `PrefsMaxScoreService` - реализация сохранения и загрузки рекорда через `PlayerPrefs`

Ядро (Core)

Основная игровая логика:

- `PlayerMovement` - перемещение игрока
- `PlayerCollector` - сбор предметов
- `Enemy` и наследники - AI поведение врагов