

Тестовое задание: Система инвентаря с Drag & Drop

Описание проекта

Данный проект представляет собой реализацию системы инвентаря для Unity, выполненную в рамках тестового задания. Основной упор сделан на чистоту кода, следование принципам SOLID и построение гибкой, масштабируемой архитектуры.

Технологический стек:

- **DI-контейнер:** Zenject
- **Анимация UI:** DOTween

Архитектура и принципы

Проект построен на архитектурном паттерне, близком к **MVP (Model-View-Presenter)**, с четким разделением ответственности:

- **Model (InventoryModel, InventorySlot):** Чистые C# классы, отвечающие за данные и бизнес-логику инвентаря. Ничего не знают об отображении.
- **View (InventoryWindowView, InventorySlotView и др.):** Компоненты MonoBehaviour, отвечающие исключительно за отображение данных и перехват действий пользователя (клики, наведения).
- **Presenter/Controller (InventoryUIManager, InventoryController):** Классы-координаторы, которые слушают события от View, обрабатывают их и отдают команды Model.

Ключевые принципы:

- **SOLID:** Проект спроектирован с учетом всех пяти принципов, в особенности **Принципа единственной ответственности (SRP)** и **Принципа инверсии зависимостей (DIP)**, реализованного с помощью Zenject.

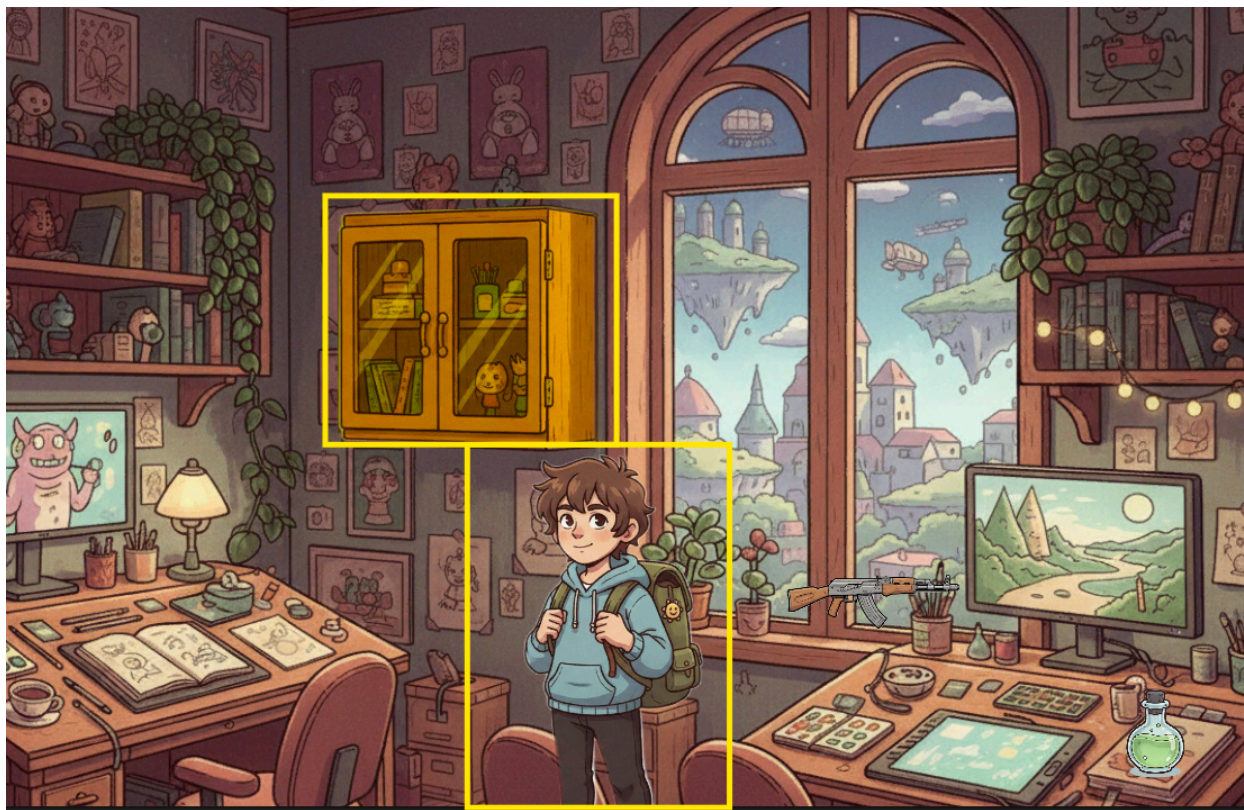
- **Разделение ответственности (Separation of Concerns):** Логика игрового мира (контейнеры) отделена от логики UI.
- **Гибкость и масштабируемость:** Архитектура позволяет легко добавлять новые типы предметов, новые контейнеры и новые механики (сортировка, сохранение), не ломая существующий код.

Как запустить и протестировать

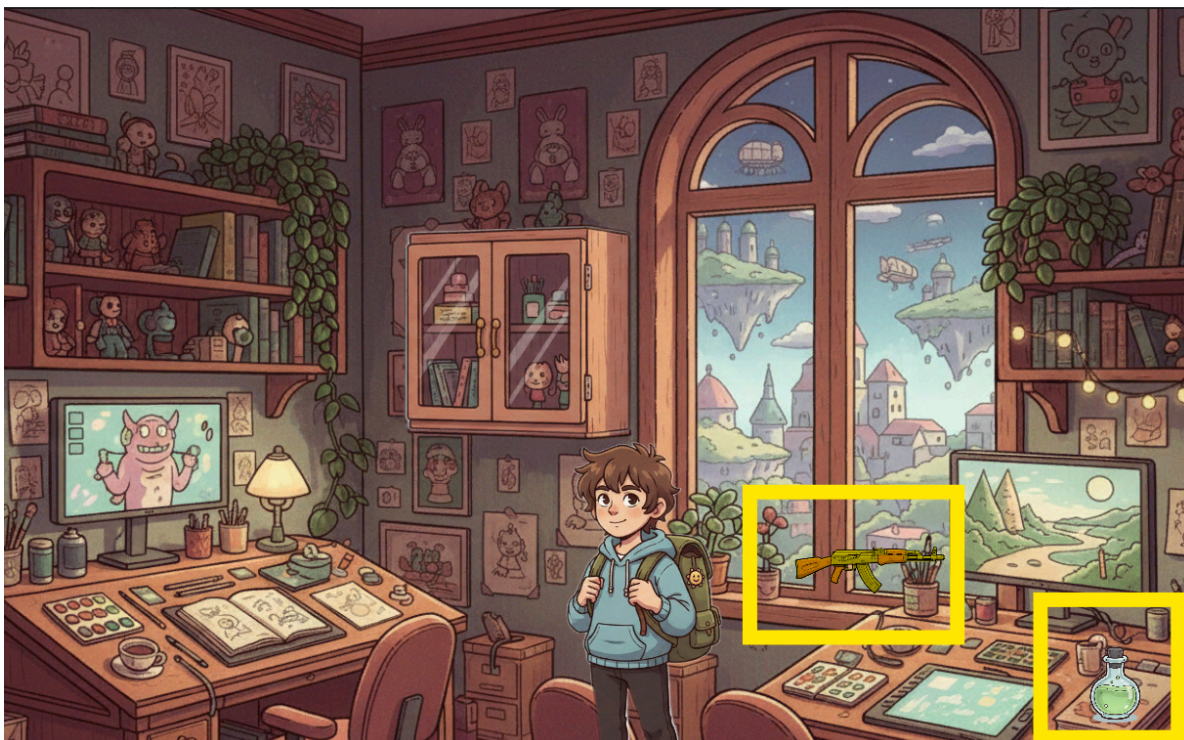
1. **Установите зависимости:** Убедитесь, что Zenject и DOTween корректно импортированы и настроены.
2. **Откройте сцену:** Перейдите в папку Assets/_Workspac/Scenes и откройте сцену **SampleScene**
3. **Запустите игру**

Интерактивное тестирование:

- **Открытие инвентаря:** Нажмите клавишу I, чтобы открыть/закрыть инвентарь игрока.
- **Взаимодействие с миром:** Кликайте на объекты на сцене ("Player", "Shelf"). Это откроет окно с двумя инвентарями: слева — инвентарь игрока, справа — инвентарь выбранного объекта.



- **Добавление тестовых предметов:** Кликните на объекты в сцене ("Weapon", "Potion"), для добавления тестовых предметов в инвентарь игрока.



- **Drag & Drop:** Перетаскивайте предметы между ячейками и между инвентарями.
- **Использование предмета:** Сделайте двойной клик по зелью, чтобы использовать одну единицу. Другие предметы использовать нельзя.
- **Удаление предмета:** Кликните по предмету правой кнопкой мыши, чтобы вызвать окно подтверждения удаления.
- **Подсказки (Tooltip):** Наведите курсор на предмет, чтобы увидеть всплывающую подсказку с его названием и описанием.
- **Сортировка:** Используйте кнопки над каждой панелью инвентаря для сортировки предметов по типу, имени или количеству.
- **Сохранение и загрузка:**
 - Состояние инвентарей (игрока и мировых контейнеров) **автоматически загружается** при старте сцены.
 - Сохранение происходит **автоматически** при выходе из игры
 - Файлы сохранений в формате .json можно найти в папке проекта: **Assets/_Workspace/Saves**.

На что обратить внимание при проверке

- **Структура проекта:** Файлы организованы по папкам в соответствии с их архитектурной ролью (Core, Gameplay, Inventory/Model, Inventory/UI/View и т.д.).
- **GameSceneInstaller.cs:** Центральная точка, где с помощью Zenject элегантно связываются все зависимости системы.
- **Разделение WorldContainer и HighlightOnMouseOver:** Пример применения SRP, где логика хранения инвентаря отделена от логики визуальной реакции.
- **InventoryUIManager vs InventoryController:** Четкое разделение ролей: UIManager — "дирижер" высокого уровня (что и когда открывать), InventoryController — "обработчик" низкого уровня (что делать при конкретном клике или драге).
- **InventoryModel.cs:** Содержит всю бизнес-логику (добавление, перемещение, сортировка), легко тестируется отдельно от остального проекта.
- **SaveLoadManager.cs:** Изолированный сервис, отвечающий только за сериализацию/десериализацию данных и работу с файловой системой.