Card War Game

A simple card battle game built with Unity where players draw cards and compete against a bot opponent. First to 8 points wins!

Project Overview

This project started as a technical assignment for a senior unity developer position - build a simple but complete Unity game that uses real-time API calls. I chose a card battle mechanic since I've always been a fan of card games like Hearthstone and Magic, but wanted something that could be built in a limited timeframe.

How to Play

1. Click "Draw Card" to begin each round
2. Your card will be drawn from a standard deck
3. The bot opponent will draw their card (with a simulated network delay)
4. Higher card value wins the round (Aces high)
5. First player to reach 8 points wins the game
6. Use the restart button to play again

Code Structure

The project follows a simple MVC-inspired architecture:

Core Components

GameManager: Controls game state, scoring, and win conditions

UIManager: Handles all UI updates, card display, and button interactions

RoundManager: Manages individual round logic and flow

DeckAPIManager: Interfaces with the Deck of Cards API to provide real cards

Helper Components

CardModel: Data model for card information and value calculations

ICardService: Interface for deck operations allowing for easy swapping of implementations

MainMenuManager: Handles menu navigation and game start/quit functionality

GameSceneUIManager: Manages in-game UI functionality like restarting or returning to menu

Technical Implementation

I ran into a few challenges during development that led to some interesting solutions:

Async API Integration

I initially used coroutines for all API calls, but switched to UniTask for better async/await support. This cleaned up the code significantly and made error handling more straightforward. You can still see some commented-out coroutine code that I left in for reference.

Simulated Network Latency

To make the bot opponent feel more realistic, I added a randomized "thinking time" of 600-1500ms. This makes it feel less like you're just playing against a simple algorithm and more like you're waiting for an actual opponent's move.

Separation of Concerns

I deliberately separated the card service implementation (DeckAPIManager) behind an interface (ICardService). This would allow easy swapping between the real API and a mock version for testing, or potentially a local deck implementation if needed.

Additional Features

Beyond the core requirements, I added:

Proper error handling for API failures and network issues

Visual feedback for the player during opponent's "thinking" time

Round result display with clear win/loss indication

Game flow management to prevent interaction during card draws

Scene navigation for moving between menu and game

Getting Started

Prerequisites

Unity 2021.3 LTS or newer

Internet connection (required for API calls)

Installation

1. Clone this repository
2. Open the project in Unity
3. Open the MainMenu scene and hit Play

Future Improvements

If I had more time, I'd love to:

Add sound effects and more visual polish

Implement card animations (flip, deal, etc.)

Create a local fallback deck when no internet is available

Add different game modes (like blackjack or poker scoring) Improve UI responsiveness on different screen sizes

Add player profiles and win tracking

Implement unit tests, especially for the game logic

Acknowledgments

Deck of Cards API for providing the card data and images UniTask for making async operations in Unity much cleaner.