Sultan's Gems

Sultan's Gems is a simple Match 3 game in which the player tries to achieve a high score within a certain number of moves by matching similar pieces. The project was created for educational purposes as part of James' **#GameInAWeek** challenge and is released freely under an MIT license. It may be of interest to those looking for a starting point in creating a Match 3 game in Unity for mobile. Bugs and feedback welcomed! @defuncart

Project Structure

This project utilizes the following project structure:

- Assets
    - Audio
    - Editor
    - Fonts
    - Imported
    - Levels
    - Localization
    - Prefabs
    - Scenes
    - Scripts
    - Sprites
    - UIAssets
- Project Settings
- UnityPackageManager

in which folders are discriptly named. **Import/DeFuncArt** contains assets which are utilized between James' various projects. This folder has a similar structure to the Assets folder, that is, containing Animations, Editor, Fonts, Other, Prefabs, Scenes and Scripts subfolders. *Scripts* contains a number of principles explained in more detail in various **#50-Unity-Tips** articles

Resolution

The game is optimized for mobile (Android and iOS), supporting the portrait orientation and aspect ratios 9:18/9:16 (phones) through to 10:16/3:4 (tablets). The UI is scaled to the desired resolution (default is 1536 x 2048).

Scenes

The game consists of three scenes *LoadingScene*, MenuScene and *GameScene*.

Modular

Sultan's Gems is designed to be modular in that new levels can be easily added by simply creating new assets and adding these assets to the *LevelManager* game object.

Game Data

Game data for each level are stored as custom assets (Scriptable objects) and loaded into memory on game load. The player's game progress is persisted to disk using binary serialization.

Coding Principles

The C# language was used throughout the game, adopting the following principles:

- Classes are named with PascalCase, i.e. MyClass
- Methods are named with PascalCase, i.e. MyClass.MyMethod()
- Properties are named with camelCase, i.e. MyClass.myVariable
- XML Documentation with the <summary> tag on all classes, methods and properties.
- Unless designated public, adhering to encapsulation, all properties are private or protected and exposed to the inspector via **[SerializeField]**.

What Could Be Improved

As Sultan's Gems was created within the constraint of a single week, there are some areas that could be immediately improved upon.

1. Presently all new board boards are instantiated when needed and destroyed when matched. An object pool would be a much better solution.
2. Presently when the game board is animating and the user presses pause, the animation continues (although is invisible). This is because all pieces are moved using coroutines on the piece's transform. A better solution would be to run the coroutines on the gameboard, with the option to 'pause' these coroutines when the game is paused (i.e. board is invisible).