

Idle Capitalist

Idle Capitalist is an idle clicker game inspired by [AdVenture Capitalist](#) in which the player invests funds into certain businesses to generate revenue. The project was created for educational purposes as part of James' **#GameInAWeek** challenge and is released freely under an MIT license.

Project Structure

This project utilizes the following project structure:

- Assets
 - Audio
 - Editor
 - GameData
 - Fonts
 - Imported
 - Localization
 - Prefabs
 - Scenes
 - Scripts
 - Sprites
- 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 9:16 (iPhone) through to 3:4 (iPad). The game is entirely UI based and scaled to the desired resolution (default is 1536 x 2048). Larger aspect ratios, such as iPhoneX or 9:18, should be playable, but not optimized for.

Scenes

The game consists of simply two scenes *LoadingScene* and *GameScene*.

Game Data

Game data for each business, manager and upgrade as 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.

Modular

Idle Clicker is designed to be modular in that new businesses, managers or upgrades can be easily added by simply creating new assets and adding these assets to the GameData game object.

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]**.