Table of Contents

Introduction

A plugin for managing game's data. Used for defining game's data structure (eg. dialogs, items, perks, units, stats), editing data and accessing this data from the code.

Creating game data file

The game data editor can open/initialize an empty file or open an existing one to try to read it and fix logical errors, if any.
Files with broken formatting or unsupported data structure cannot be opened.

To create new game data file, do following:

1. Right-click inside the Project window in your Unity editor
2. Select Create → Game Data → Game Data (<format>) in the popup menu

Launching data editor

To launch editor do following:

1. Open **Project** window by using **Window → Project** menu
2. Locate your game data file (it should have .gd.json extension).
3. Double-click the file.
4. The game data editor window should appear near **Game** window

Setting up data structure

The editor structures data as documents consisting of elements, each having a name and a value. Values can be either numerical or string. Other documents maybe be used as values as well.

Each document's structure an purpose are described by another special document called Entity. Each

entity consists of attributes and additional parameters.

Data-describing data is called metadata. For now you will need to reload your editor page each time metadata changes (we will fix this in the following updates).

For example: if you need to store information about game characters, you create an entity named **Character** and use attributes **Name**, **Gender** to describe which parameters your characters will have.

Editing documents

Every document you will encounter in the editor belongs to some entity.

You can see the list of the entity's documents by clicking its name in the left menu of game data editor.

Creating

To create a new document, click Create button just above list of documents.

Editing

To edit a document, double-click it in the entity's document list, or select it from the list and click Edit.

Deleting

To delete one or more documents, select them from the list and click Delete. You can also click Delete while you're editing a document to delete it.

Generating source code (C#)

Game data editor can generate API for accessing game data.

1. Open Project window by using Window → Project menu
2. Select your game data file in Project window.
3. Open Inspector window by using Window → Inspector menu
4. In Inspector window: select Code Generator → C Sharp
5. Set up your code generation parameters.

Accessing game data (C#)

Accessing game data from source code is possible by using pregenerated API classes.

Loading data

The following C# code creates API class and loads your data.

```
var fileStream = File.OpenRead("gamedata.json");

var gameData = new GameData(fileStream, GameData.Format.Json);
```

The loaded data is read-only.

Accessing documents

You can access your documents as a list:

```
var characters = gameData.GetCharacters() // -> ReadOnlyList<Character>

var characters = gameData.GetCharacters(onlyRoot: true) // -> ReadOnlyList<Character>
```

Or you can access specific documents by their Id or 'unique' fields:

```
var character = gameData.GetCharacter(characterId); // -> Character

var character = gameData.GetCharacterByName(characterName); // -> Character
```

Settings entities are accessed by name:

```
var resetTime = gameData.LootSettings.ResetTime; // -> TimeSpan
```

Formulas

Formulas are executed with Invoke method:

```
var reward = gameData.LootSettings.RewardFormula.Invoke()  // -> int
```

Formula parameters are passed as common arguments of method.

Advanced Topics

Futher infomation could be aquired from online documentation.