QuickSheet specification

By Nichos

2016.12

Requirements:

A good data layer solution should satisfy these requirements:

1 The data files are friendly to designers. We choose Excel format.

2 The exported files are friendly to the game. It means it should be:

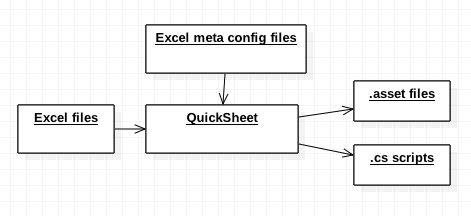
1. Unreadable. To prevent crack or make crack hard.
2. Compact size. Since the data files will be used for live update, so the file size should be as small as possible.
3. High performance. It’s fast to load and parse, cost low memory usage.

Based on these requirements, popular xml or json is ruled out. We choose .asset format. It’s a native Unity format, binary, easy for serialization.

3 It can auto generate script files. Since the data layer scripts are basically boilerplate code, it’s completely feasible to auto generate them. The only thing we should do is config some meta settings for each excel sheet, then press a button, all script files are generated.

Solution:

We choose QuickSheet since it satisfies all requirements above. The graph shows what QuickSheet does:

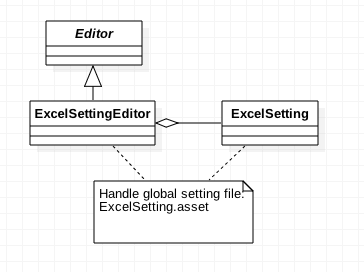


QuickSheet code analysis:

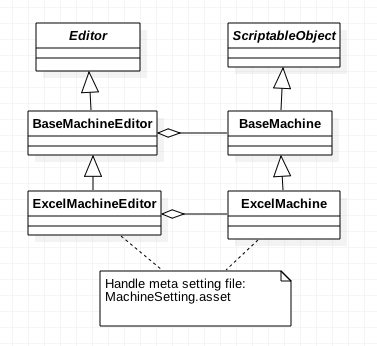
1 There are two config files:

1. one global setting file: ExcelSetting.asset
2. some meta setting files for every excel sheet format: Ex. MachineSetting.asset

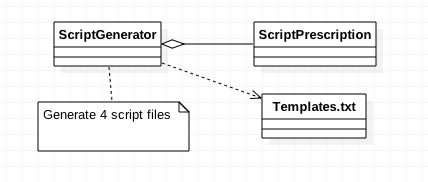
The class graph related to global setting file:



The class graph related to single meta setting file:



2 The class graph of script generator:



It generates 4 .cs files, 2 for editor use, 2 for runtime use. For example, when the sheet name is “Reel”, the generated files are:

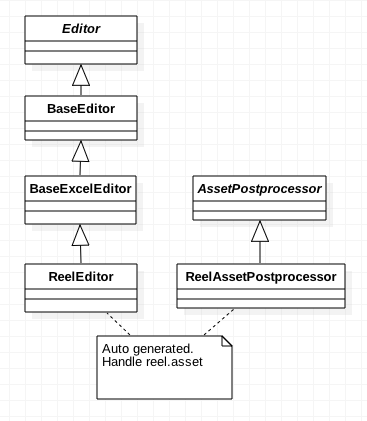
For editor:

1. ReelEditor: define how reel.asset shows
2. ReelAssetPostprocessor: define how Reel sheet data is exported to reel.asset

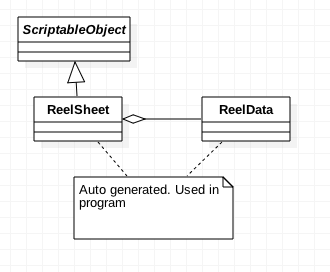
For runtime:

1. ReelSheet: corresponding to the sheet “Reel”
2. ReelData: corresponding to one row in the sheet “Reel”

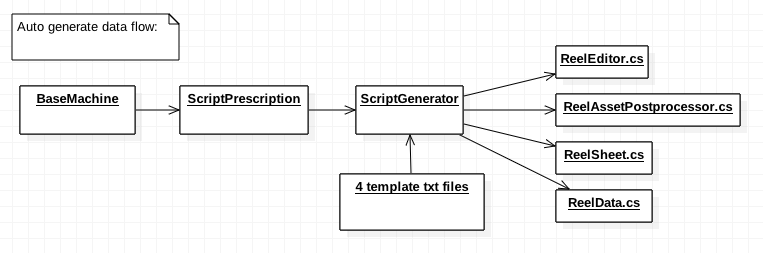
The class graph for 2 editor files:



The class graph for 2 runtime files:



The overall auto generation data flow is:



Improvements  
We did some customization and improvements in QuickSheet code:

1. remove google sheet support
2. support more data to config, ex. export path
3. better naming for exported files, class names, variable names
4. support ColumnHeader serialization and deserialization
5. ignore column header of the name with prefix \*

And more is possible in the future.

Use guideline

1 The auto generated code should be all under the directory /AutoGenerated for maintenance convenience.

2 The generated data class such as “ReelSheet” is corresponding to the format in excel files. But sometimes the data class is not straightforward or easy to use in program. So it needs further transformation or second process based on the exported raw data.

To do that, we shouldn’t modify any auto-generated code. Instead, for every exported “XYZSheet” data class, we create a new wrapper named “XYZConfig”. It contains a “XYZSheet” instance and all new interfaces we need to add. The client outside should only depend on the wrapper class.

The graph shows:

