

Table of Contents

Best Practices	1
----------------------	---

Best Practices

VASSAL is a powerful toolkit. As a result, there may be any number of ways to accomplish a particular goal, all of which may be equally correct. It's not possible in this guide to outline all the methods for accomplish a given objective.

There are, however, a set of best practices gleaned from other VASSAL designers that can save a lot of time and effort.

The following guidelines may be helpful to keep in mind when designing a module or extension.

Know the Game

Get to know the game before creating a module. You should review the rules in detail even if you've played the game itself many times, looking for potential issues that can slow down the module design process. How does the game proceed? Are there any special rules that standard VASSAL tools can't handle and may require custom coding?

Don't Try to Enforce the Rules

VASSAL is not, in general, intended to enforce rules. Rules enforcement should be left to the players, just as it would be at a tabletop. Is it possible to make a module track each piece's movement points, and types of attacks, and attack results? Probably. Is it a good idea? No. The process will probably involve much work on your part, for little advantage. In addition, VASSAL may not include the tools you need for proper rules enforcement. A tabletop game proceeds by player cooperation and mutual knowledge of the game rules; take your cue from that.

Have All the Required Files on Hand

It's a good idea to collect all the art, text, and other files you're going to use in the module *before* embarking on the creation of a module. The creation process will go much more smoothly if you have all the files you'll need ahead of time. The art can be gathered by scanning the original artwork, you can create your own from scratch, or use some combination of the two. You can create text and help files in any text or HTML editor.

Limit Automation

There's no denying that automation can be a powerful tool for making gameplay easier. Global Key Commands and Trigger Actions are excellent methods of automating nearly any standard game function. Bear in mind, however, that VASSAL is intended to mimic the experience of sitting at a table with your opponents. As a result, the less automation in the game, the better. During game play, too much automation reduces the flexibility players have, as decisions may be made for them automatically.

In addition, a highly automated scheme can make it harder to track down logical errors and game

misfires during the module development stage. Another issue could be the computational time required for automated tasks to execute. In most cases, automation is best used for bookkeeping or repetitive tasks.

Programmatic Efficiencies

Gameplay can be made more efficient and less time-consuming by taking advantage of VASSAL's features. For example, many board games include a turn track, which is an actual physical space on the game board where a turn tracker piece is moved. Use of VASSAL's turn counter can easily replace this, saving you the trouble of creating the turn track window and graphics. Status markers that require placement on top of other counters can be simulated by the use of the Layer trait, without having to drag markers to the board.

Documentation

Players always welcome documentation on how to use your module. Although they may be well-versed in the boardgame version of the game you've designed, a VASSAL module may present unique challenges when it comes to gameplay that need additional explanation. What may be obvious to *you* may not be so obvious to players.

You can use the **Help** menu to add your own designer's notes or getting started guides, explaining module concepts. If the game or module is particularly complex, consider adding a Tutorial to show how the module is intended to work.

Best Practices: New Versions of Existing Modules

Play a Game!

Test out your module before you publish it by playing some full games. You may realize you've forgotten about a rule or consideration that may only come to light through gameplay. Professional game designers insist on rigorous playtesting, and you should too.

Learn from Others

There is no need to reinvent the wheel. With nearly one thousand VASSAL modules available, chances are high that a game similar to yours has already been turned into a module. In fact, most module designers got started by examining the workings of existing modules and learning from them. Previous module designers may already have solved some of the issues that your game presents. With modules easily available for view in the Module Editor, why not examine previously used methods and learn from them?

The VASSALengine.org site also features a forum for asking module design questions.