Examining Modifiability and Modularization in JSettlers

Author: Hannah Raju (V00797276)

Date: 15 February 2019

Details: SENG 480 - Assignment 1

Table of Contents

JSettlers is a web-based application based on Settlers of Catan, a popular multiplayer strategy board game. JSettlers allows users to play Catan online against other users or AI. The game involves collecting and trading resources to construct and expand your existing settlements.

To play JSettlers, users must download the Java Runtime Environment then follow the detailed instructions in the documentation to set up a server. A new game can then be setup according to various options, such as the number of players and number of Victory points required to win [1].

1. Introduction to JSettlers

In this section, the modularity and testing procedure of JSettlers is explored. The resulting modifiability of the system is analyzed. The assumption is made that the developers of JSetters want to ensure the system is up to date with all released expansion packs for Settlers of Catan.

The structure of the game was determined by examining the source code. Like its parent game, JSettlers is structured such that an entire expansion can be added with little difficulty. This is due to its modularization with respect to separate elements of the game. Each class of elements is broken into subclasses extending from their respective parent class. This tree structure ensures simplicity when added new game element.

The time to add an expansion pack in its entirety (including components, behavior, and new scenarios) to JSettlers is estimated to be at most three hours. However, the current documentation for JSettlers is very minimal and no standard testing practices or metrics are described. This estimation is based upon analysis of the source code.

2. Construction of Quality Attribute Scenario

This section outlines the process of analyzing the system to construct the use case quality attribute scenario and the growth scenario. Modifiability was chosen as the focus quality attribute for JSettlers because of its highly modularized structure. Table 1 provides the details of the use case quality attribute scenario for modifiability.

Table 1: Use Case Quality Attribute Scenario for Modifiability

Aspect	Details
Scenario Name	Adding new expansion or game mode of Catan to JSettlers
Business Goals	Ability to add, test, and commit new game mode efficiently
Quality Attribute	Modifiability
Stimulus	New game mode developed and want to add it to the system
Stimulus Source	Catan creators introduce new expansion pack
Response	New game functionality integrated with existing game structure
Response Measure	Changes and testing took at most 3 hours (for one expansion pack).

The release rate of new Catan expansion packs is approximately once every three years [2], so the time required to keep the system up to date (in terms of expansion packs alone) is very minimal, requiring roughly three hours every three years. The time required for other general maintenance is not considered here.

While changes may require manual propagation through the system, where the changes must occur is very straightforward. Figure 1 offers a noncomprehensive overview of the structure of the game. With the current structure, adding new game elements is simple and efficient. The intuitive organization of the various game components contributes to its simplicity. For each individual component being added, only three significant changes are required: defining the class containing the component's behavior, implementing the component's effect on the overall state of the game, and adding reference to the element in its respective module. For example, if the added component effects the 4-player board, appropriate changes are made to the 4-player board component and must propagate up to the parent board module. This ensures, in theory, that entire expansion packs can be added to the system's functionality relatively efficiently.

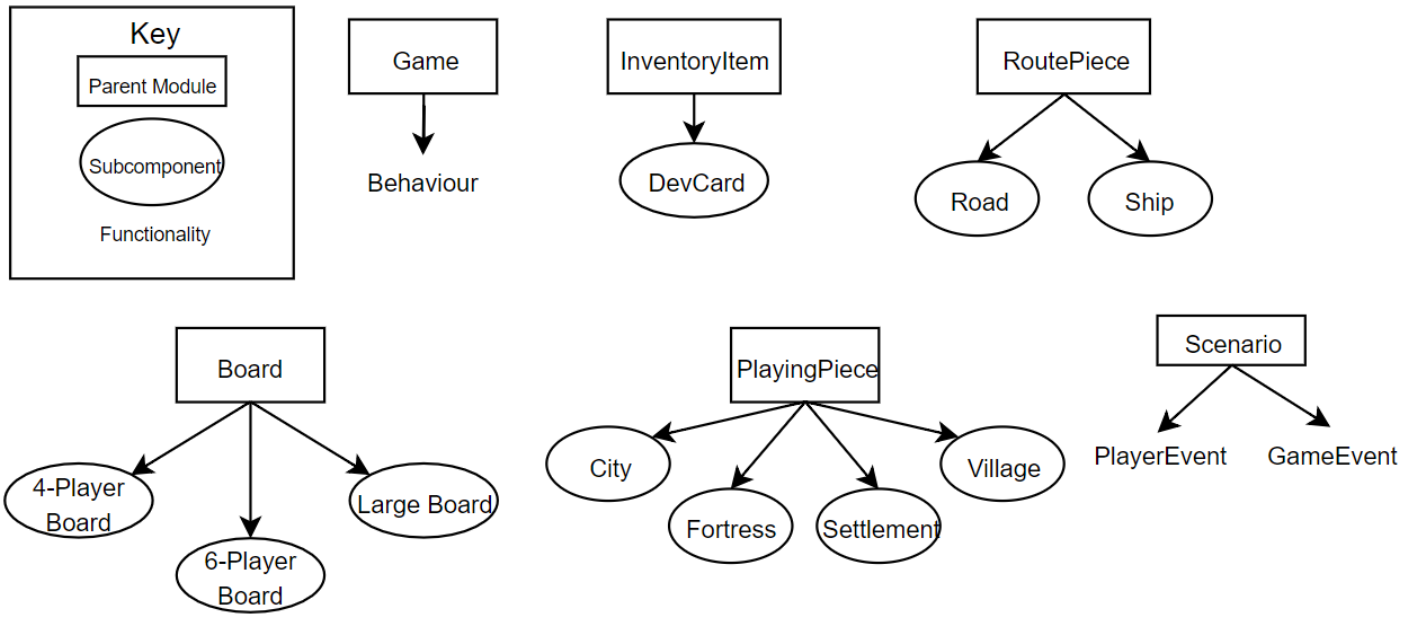

Figure 1: Use-Case overview of JSettlers game structure

3. Expanding QAS for Growth

In this section the use case QAS defined in Table 1 is extended to create a growth scenario. The growth aspect of JSettlers to be analyzed is the hypothetical case that Settlers of Catan begins releasing expansion packs at a rate increased from one every three years to ten per year. Under the assumption stated in the Introduction, JSettlers strives to ensure their system is up to date with all released expansion packs. An additional assumption is made, to demonstrate the worst-case scenario, that the ten expansion packs are released at once, thus they are added to JSettlers around the same time. The derived growth scenario is described in table 2. The bolded text identifies which aspects of the new growth scenario differ from the original use case QAS.

Table 2: Growth Scenario for Modifiability

Aspect	Details
Scenario Name	Adding new expansion or game mode of Catan to JSettlers
Business Goals	Ability to add, test, and commit new game mode efficiently
Quality Attribute	Modifiability
Stimulus	New game modes developed and want to add it to the system
Stimulus Source	Catan creators introduce new expansion packs
Response	New game functionality integrated with existing game structure
Response Measure	Changes and testing took at most **5 hours per expansion pack**.

The use case response measure defines a modification time of approximately three hours per expansion, while the growth scenario defines a modification time of approximately five hours per expansion pack. This discrepancy arises when potential interacting functionality and behavior of simultaneously added expansion packs are considered. Therefore, the addition of ten new expansion packs would, in theory, take approximately 50 hours of work; very little time over the course of an entire year. Figure 2 demonstrates how new components could be integrated with the existing structure of JSettlers. It is evident from the modularization that the structure of JSettlers allows for frequent, robust additions to its system. Consider for example an entirely outrageous release rate of Catan products, 100 per year. This would require an estimated 500 hours per year (only about 10 hours per week) to keep the system updated.

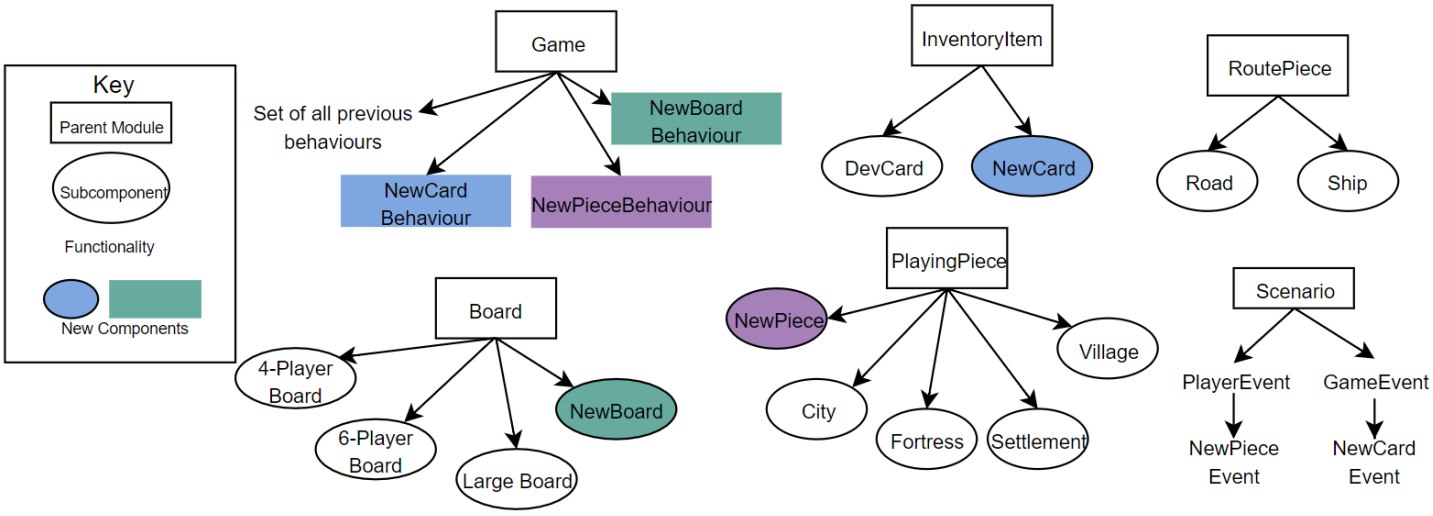

Figure 2: Hypothetical growth scenario for JSettlers game structure

4. Conclusion

Overall, the current structure of JSettlers is well-suited to withstand the frequent modifications required to keep the game up to date. With the current release rate of one expansion pack every three years, JSettlers is required to perform an estimated 3 hours of updates and testing every three years. Considering the growth scenario, in which Settlers of Catan begins releasing expansions more rapidly, only around 50 hours per year is required. While the source code itself does not explicitly contain anything to aid in more frequent additions, the structure of the system lends itself to efficient modifications.

5. References

[1] http://nand.net/jsettlers/devel/

[2] https://en.wikipedia.org/wiki/List_of_The_Settlers_of_Catan_products