The Iterator Pattern in the Solitaire game

Björn Forsberg (Group 13)

May 9, 2014

The iterator pattern provides a way to iterate over data structures without taking the data structure representation into consideration by acting as an abstraction layer between the calling code and the data structure. The abstracted representation of the data structure is called a Collection.

The current implementation of the Solitaire view would benefit from the Iterator pattern, since it is now hard coded to use lists. Should we decide to change this representation, the whole view part of the program has to be changed. Since the most important functionality of the view is to draw all the cards in stacks, and stacks are just Collections of cards, the iterator pattern could iterate over the stack (sequentially) and print the cards.

The design pattern is implemented in such a way, that the calling class that wishes to iterate over an object calls a function named something like `createIterator()` on the class. This operation will create an iterator which the calling class can use to call functions like `next()`, `prev()`, `has_next()`, and other functions used to iterate over the collection.

Internally, a collection implements the collection interface, so that other classes can access these functions in a coherent way. The iterator for the specific collection is included within this outer class, and inherits from a iterator interface, so the specific implementation within this collection can be ignored, and the common interface used instead.

The reason this design pattern is not implemented at this point is that the focus of the development has been on the graphical and undo parts, and from this viewpoint, the data structures used are only needed to be able to display something, and how they are implemented is secondary. When the required functionality is implemented, it would be interesting to see if the design pattern could be implemented.

References

[1] *Iterator Pattern - Object Oriented Design*, accessed 8 May 2014. [Online]. Available at `http://www.oodesign.com/iterator-pattern.html`.