HACETTEPE UNIVERSITY DEPARTMENT OF

COMPUTER ENGINEERING

BBM 203 ASSIGNMENT 1



Name Surname – Number

Subject : Arrays

Programming Language : C++

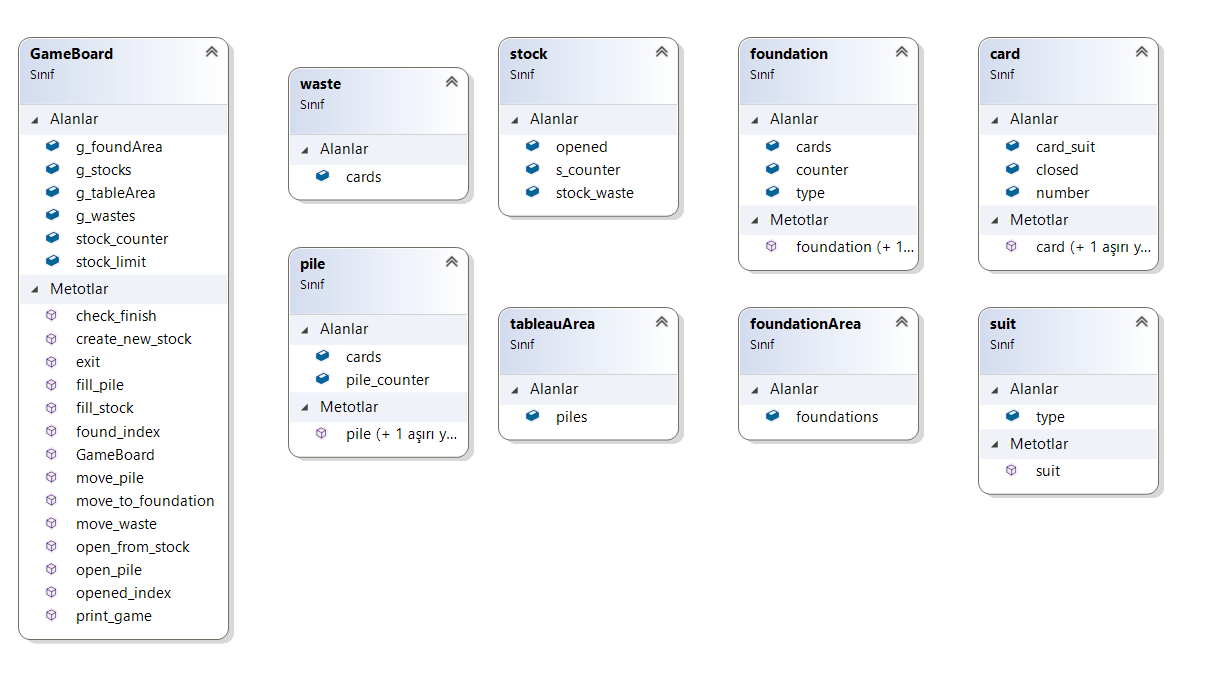
1. **Defining problem**

Coding array based solitaire game in C ++ language. Using OOP (object oriented programming) while coding. When using OOP, efficiently managing the classes we use to design the game and efficiently managing the arrays within the classes.

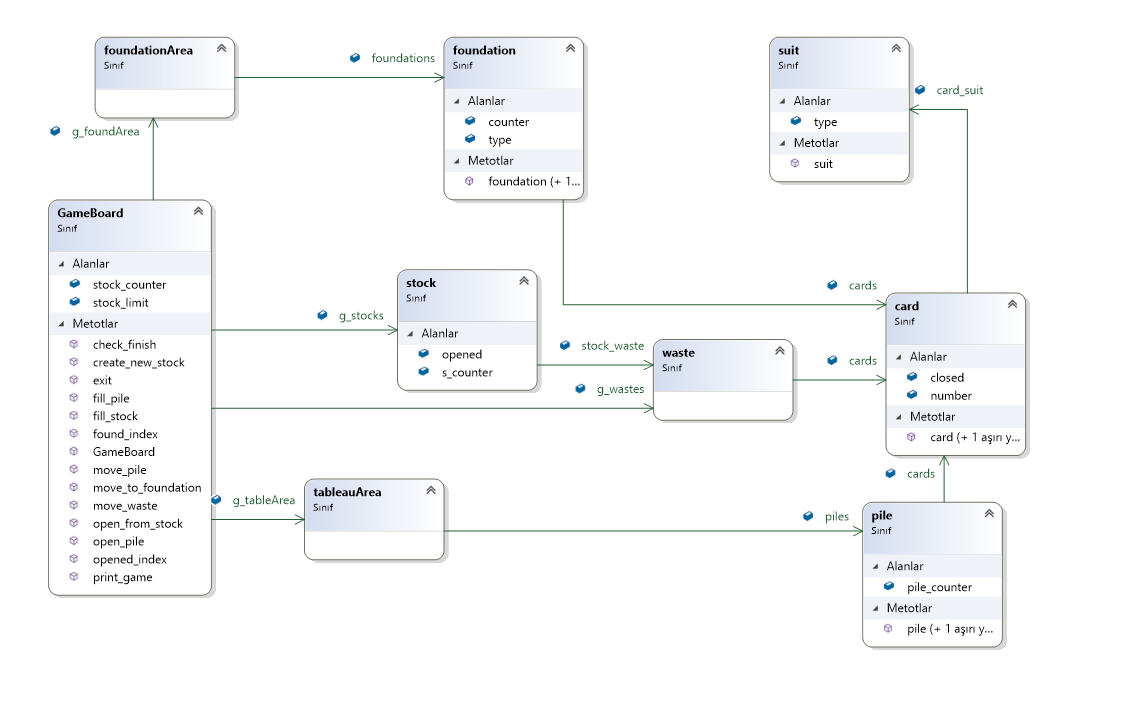
1. **Explanation of your approach**

I took care to keep the main () function short and understandable. When the program is run, the game (GameBoard class) object is created first. In the constructer of the GameBoard class, the basic objects of the game are created and the necessary fields (allocation with new function) are allocated. Secondly, ofstream creates an output file depending on the argument. then, with the create\_deck function, the cards in the input file are properly inserted into the game object. With the print\_game () function in the game, the initial state of the game is written to the output file. then the file is not read according to the argument with the apply\_command () function. When each line is read, necessary actions are taken, the results are written to the output file. Finally, with the check\_finish () function in the game, it is checked whether it has won the game. After all operations are finished, the output.close () file is closed, the areas reserved for the game object are freed with the function allmemory\_free (). With the delete game code, the object is deleted and the program is terminated.

1. **Explanation of class diagram**



**Association**



1. **Explanation of class arrays**

While creating the class diagram (while coding), I based on the picture you gave in pdf (Fig.1: Klondike Solitaire Terminology). Gameboard is an inclusive object. Other classes are included in Gameboard.

TableArea has 7 piles. every pile has cards. each card has a suit.

The Stock object has an 8-unit array called stock\_waste. Each array has cards. each card has a suit.

Waste is initially 3 cards blank. Depends on Stock.

foundationArea has 4 foundation. All foundations are blank. All foundations have arrays called cards. These arrays have cards. each card has a suit.