OBJECT ORIENTED ANALYSIS DESIGN

TCP2201

GROUP NAME: CODEBREAKERS

PREPARED BY:

1.	CHAI YING HUA	1141328508	011-23175127
2.	RAMADAN ALI HOUSSEIN	1141123660	-
3.	SHUBAR ABDUELHAKEM.G.ABDUSALAM	1141327949	018-2304569

SECTION: TC02 TT05

PREPARED FOR:
DR. IAN CHAI

REPORT SUBMISSION DATE:
19 SEPTEMBER 2016

Introduction

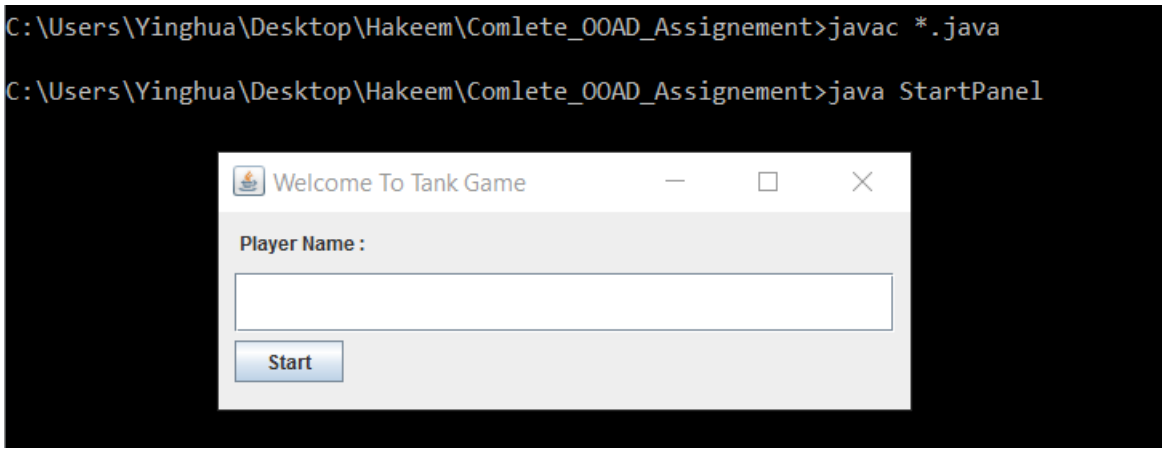
This documentation describes the details of OOAD tank games, guiding the user how to use the program and explains use-case diagrams, sequence diagrams and class diagrams. The design patterns used in this project will be explained and shown.

Acknowledgement

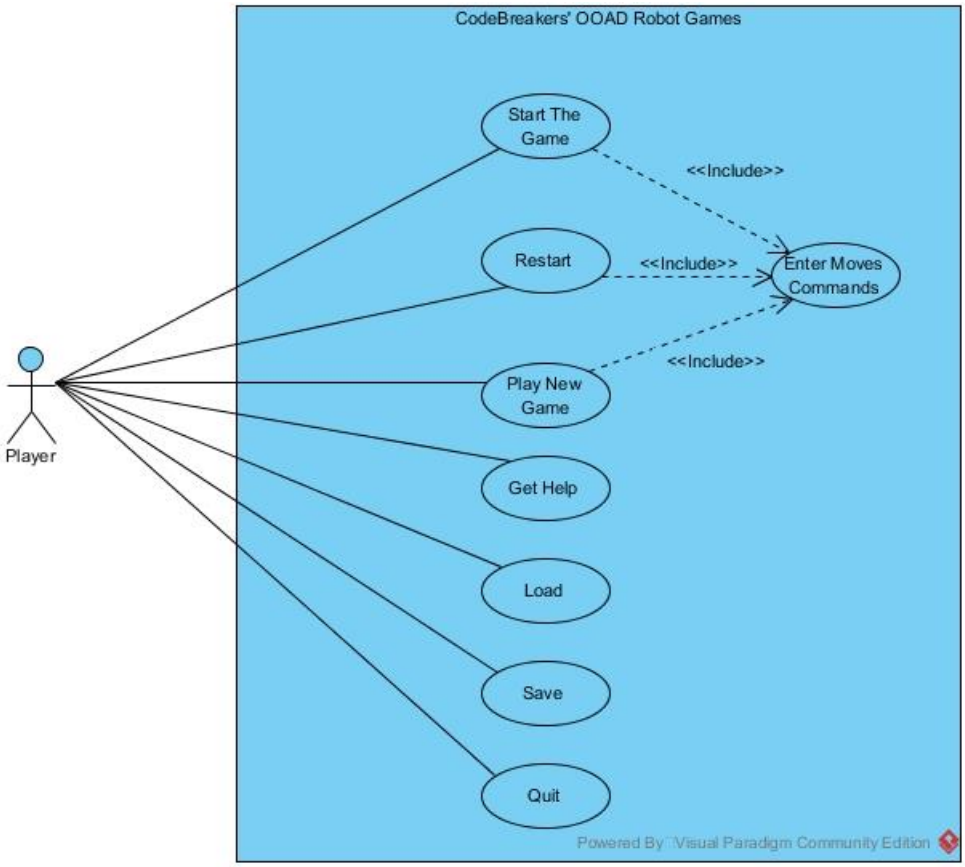
First of all, we are grateful and thanks to our lecturer **Dr. Ian Chai** who give us a very details good lecture and willing to guide and help our assignment. Moreover, we appreciate the teamwork and good task-management between the teammates in order to finish the whole assignment. Last but not least, we are sincerely thank to our classmate and friends for their guidance and enlighten to finish this projects.

Compile the file

This game is recommended compile with commands line.



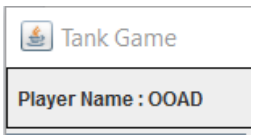
1. Use Case Diagram:



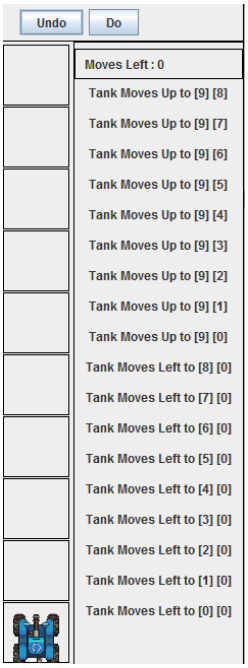
The use case diagram demonstrates how the player interacts with the game by showing the functions player may use while playing the game.

a. Start the Game:

The user is able **to start the game** by pressing the [Start] buttons, after filling their names. The player name will be uses as the filename to **save the sequences** and **load the sequences**. The player name will be set in the game.



b. Enter Move Commands: The user requires inputting the key to form a user sequence in order to make the tanks move. The undo buttons allow user to undo and modify their moves, the do buttons allow user to execute the move sequences and order the tank to animate on the screen.



c. Restart:

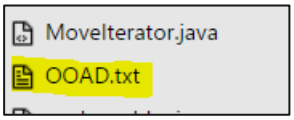
The user can challenge the new computer sequences by pressing the [Restart] button on the menu. All the user sequences and computer sequences will be cleared; the computer ArrayList will fill the new computer sequences.

d. Play New Game

The user can re-challenge the game with the previous computer sequences by pressing [New Game] buttons in the menu. The computer sequences remains but the user sequences are cleared.

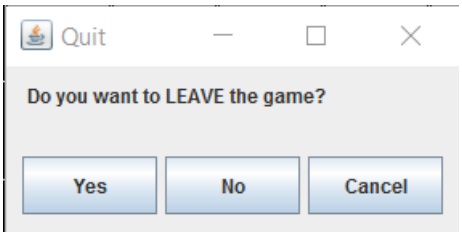
e. Get Help : The user can get the guide by pressing the [Help] buttons.

f. Save : The user can save both human and computer sequences according to their player name. Both sequences will be saving into the file with the player names.

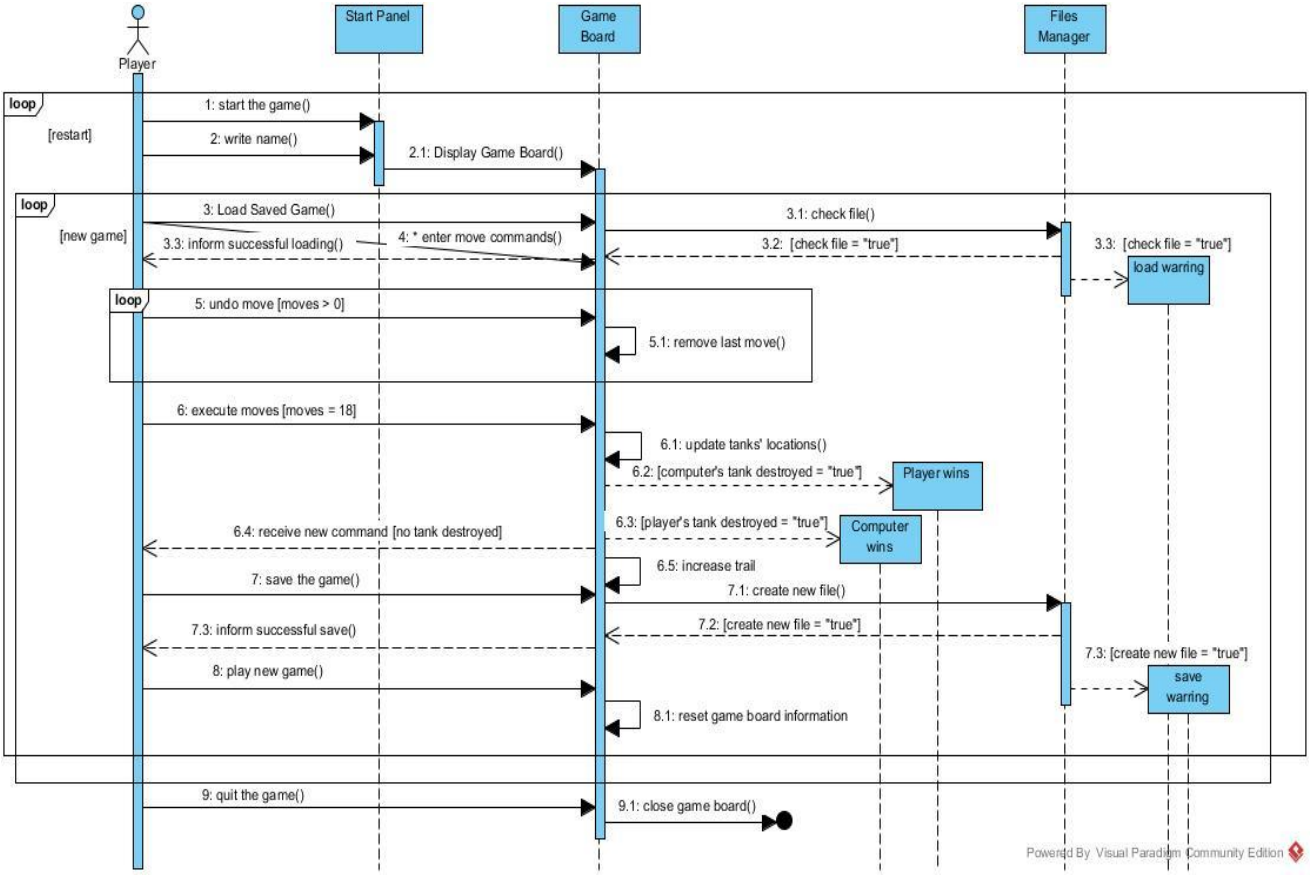


g. Load: The user can load the sequence base on the player name. If the player name match the file name, both sequences will be loaded at the same time. The user can replay how the scenario last time they play, and press [New Game] to challenge the computer sequences loaded.

h. Quit : The user can leave the game by pressing the [Quit] Buttons. A message will display to make sure the choice of the user.

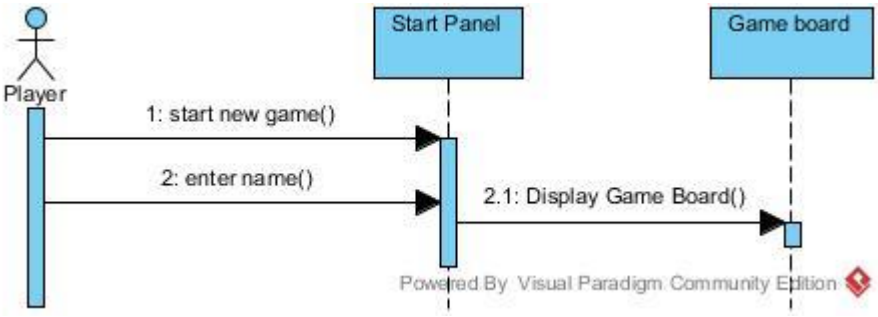


2. Sequence Diagram:



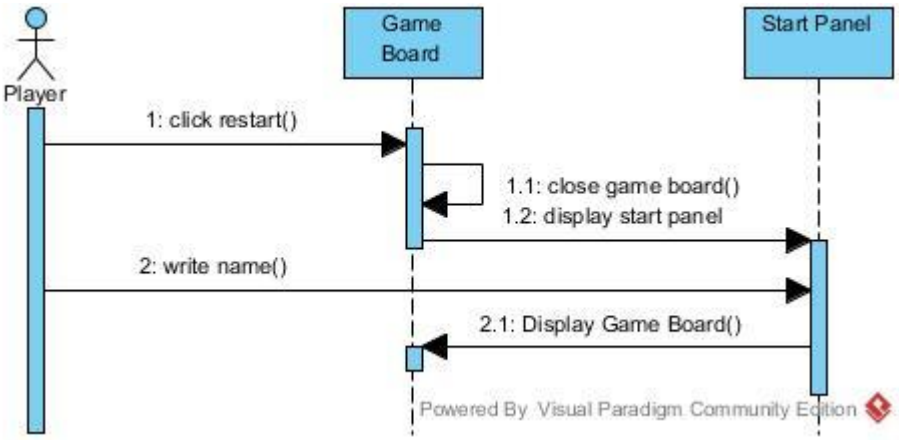
The above sequence diagram shows how the player uses the game functionality over time sequence and how the use cases can be used in the given life time.

2.2 Start The Game:



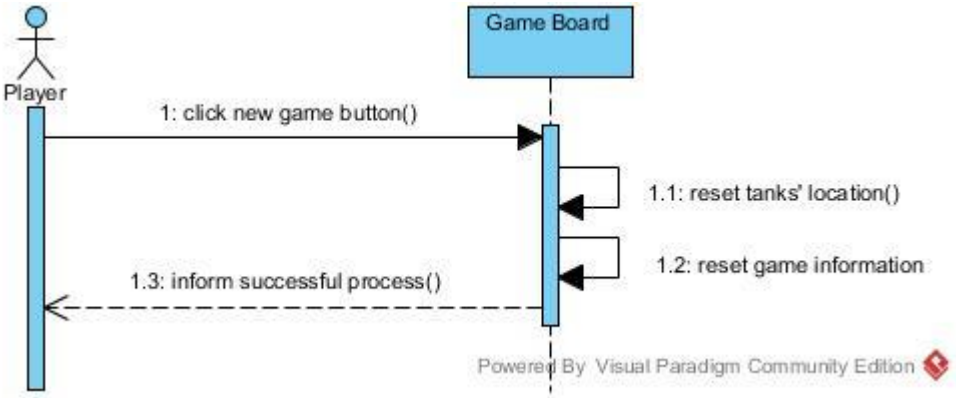
When the user start the game the start panel will come out with text box where he/she can write his name. After writing the game and clicking start button the game board appears and player can enter the commands.

2.3 Restart



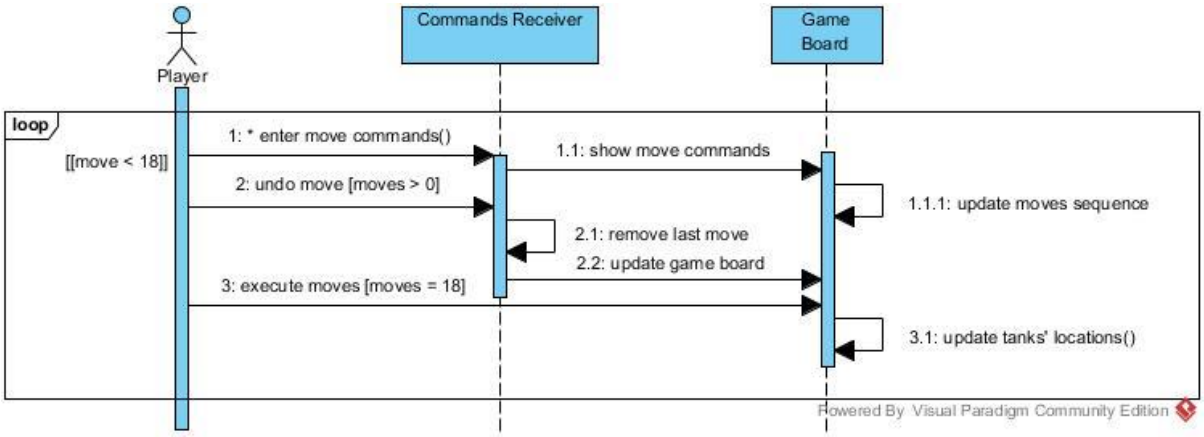
After clicking restart button the current game board will be closed and the game the start panel will come out with text box where he/she can write his name. After writing the game and clicking start button the game board appears and player can enter the commands.

2.4 New Game



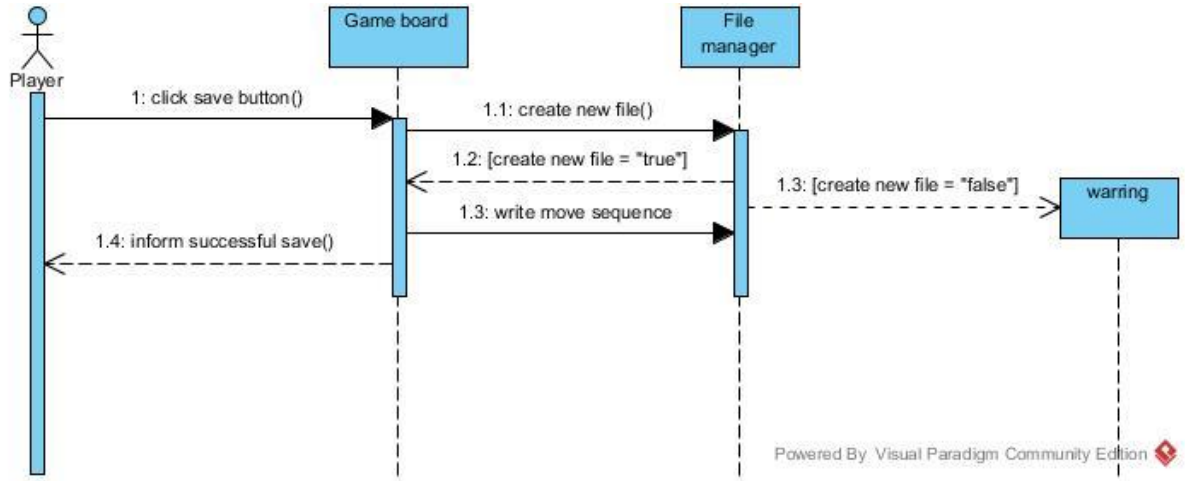
In case the sequence of moves entered by the player did not destroy any of the tanks, the player in asked to enter new sequence of moves by clicking new game button that in turn will reset the tanks' locations.

2.5 Enter Commands



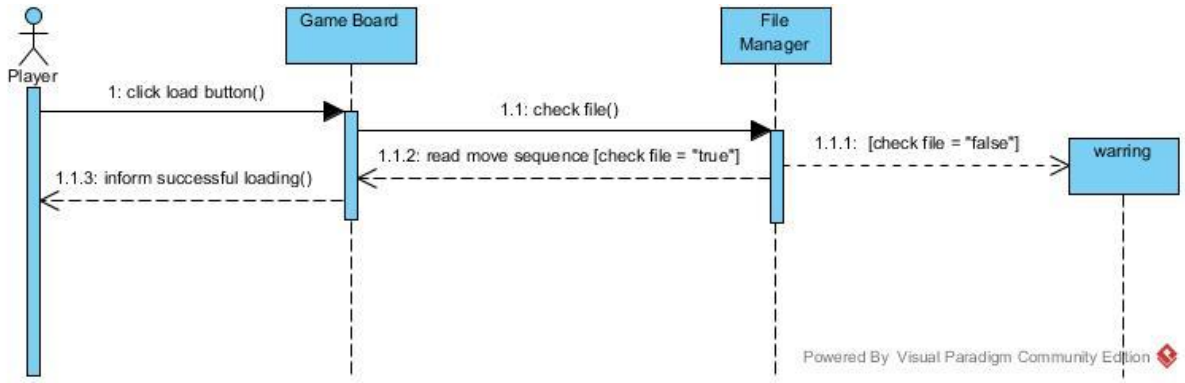
The player is entitled to enter 18 moves and each move will be shown in game board. Undo button provide a functionality to the player to delete the last move while the do button will execute the 18 moves entered and update the game board and tanks' locations.

2.6 Save



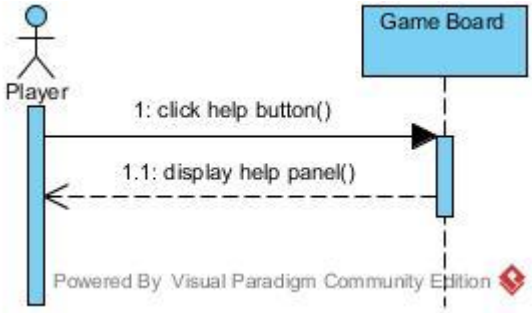
By clicking on save button a new file with the name of player entered at the beginning of the game will be created and the last moves sequence of the computer and player will be written in the file. In case the file cannot be created an error message will come out.

2.7 Load



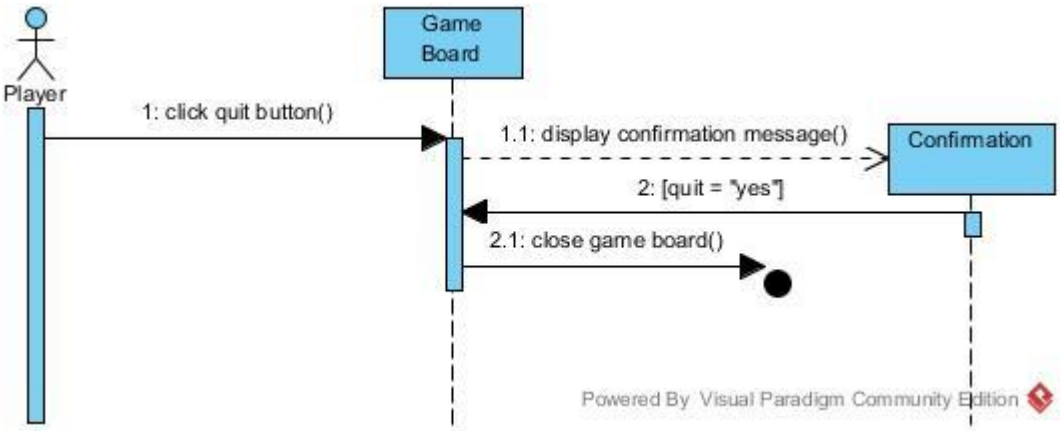
By clicking on load button a file with the name of player entered at the beginning of the game will be opened and the moves sequence of the computer and player saved in this file will be read to the game. In case the file cannot be found an error message will come out.

2.8 Get Help



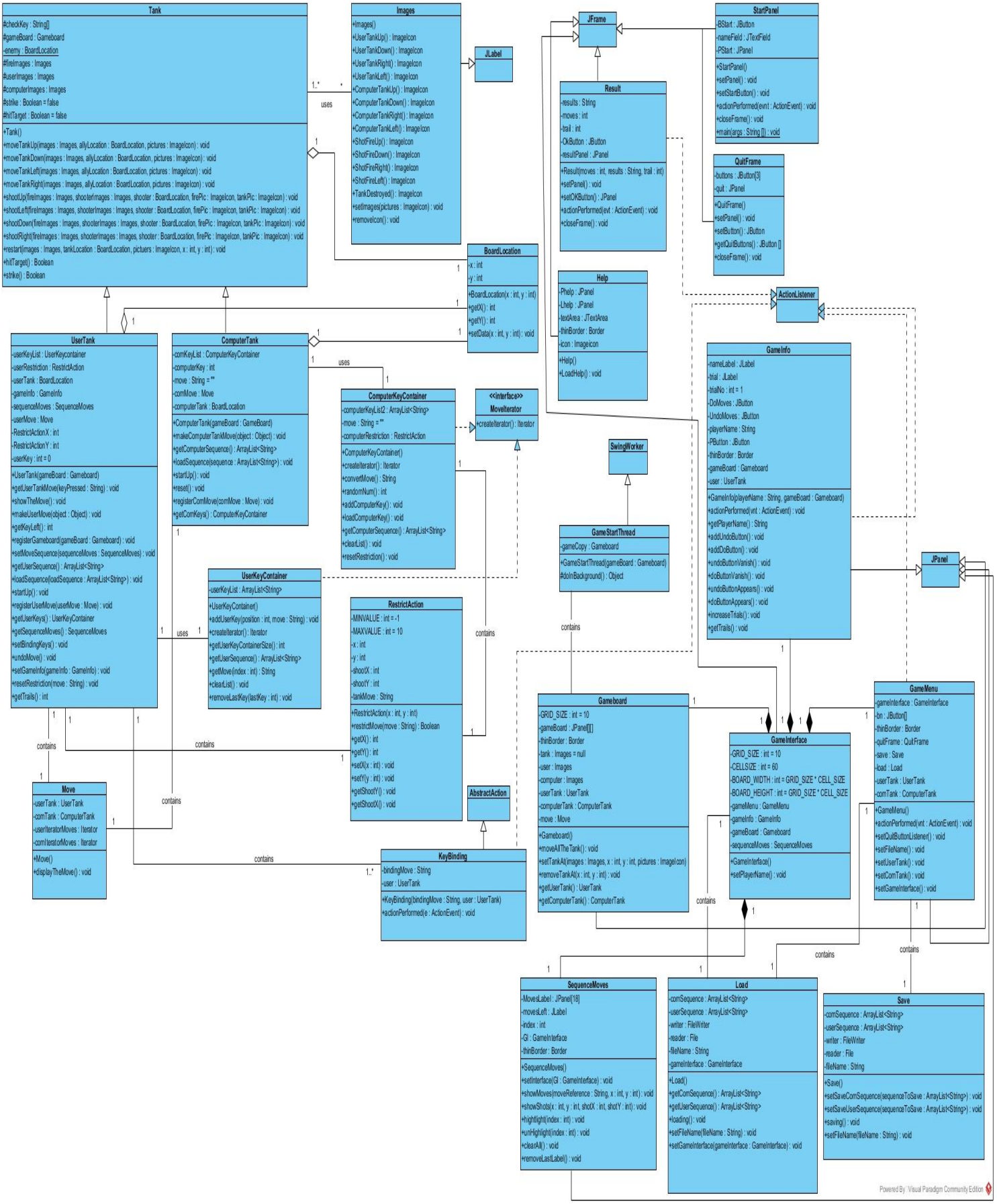
Help instructions can be obtained by clicking on the help button on game menu.

2.9 Quit

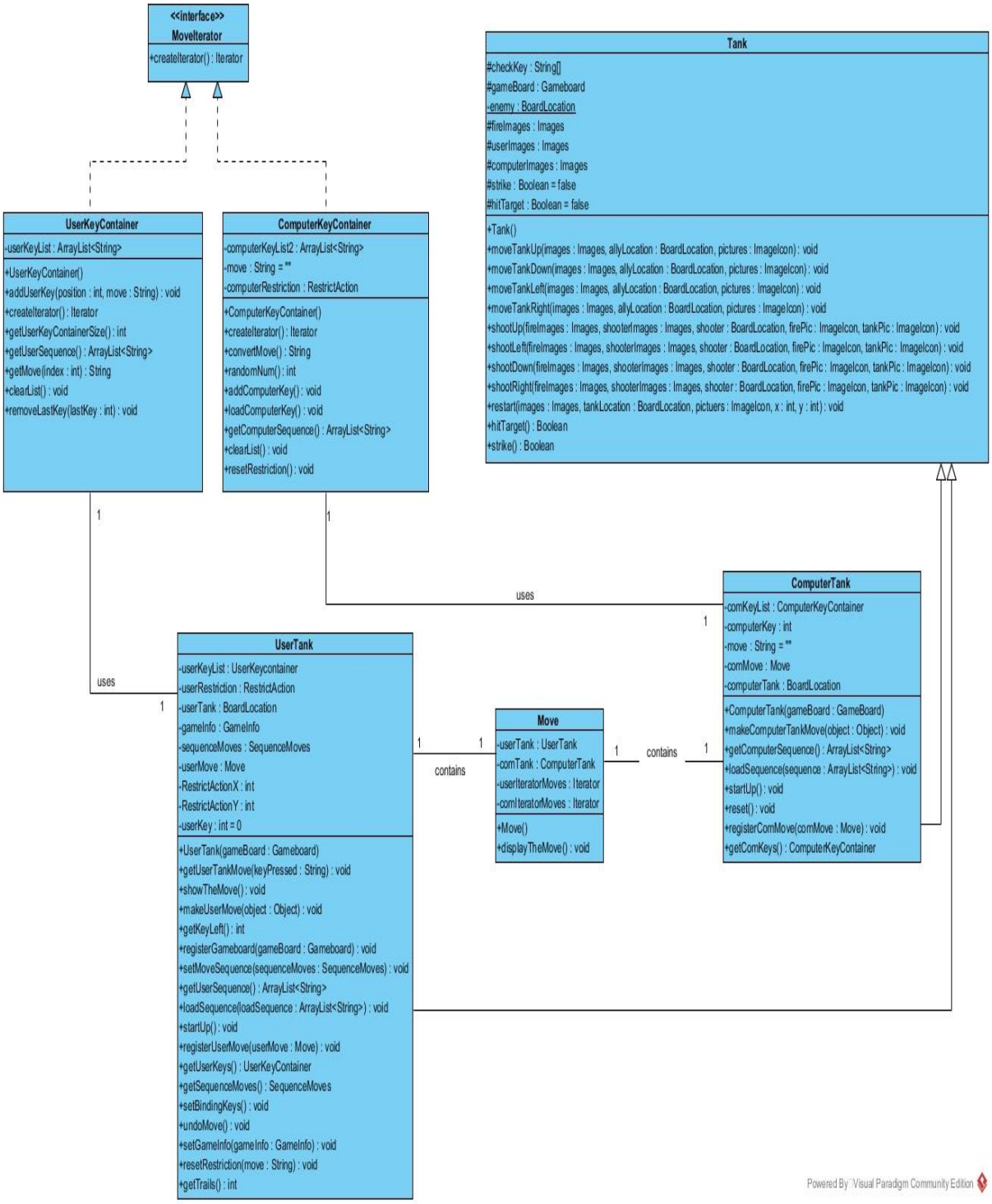


Quit button in game menu will make the game end. However, a confirmation message will allow the player to confirm his/her decision.

3. Class Diagram



3.1 Design Patterns



Iterator design patterns.

The iterator design patterns required concrete container class, iterator interface and clients. From the diagrams above, UserKeyContainer class and ComputerKeyContainer class are the container class that implements the MoveIterator interface. Both container classes contain Array List that store the move sequences of both tanks. The Move Iterator interface contains methods that allow client to create iterator in the container and use it.

The Move class, Computer Tank class and UserTank class are the clients that use the container class to access the elements of an aggregate object. The ComputerTank and UserTank class create an instance of container class to allow human and computer to add the sequences into the array to store all the objects. The move class create the iterator within the ArrayList obtained from ComputerTank and UserTank and loop through each objects to access the elements.