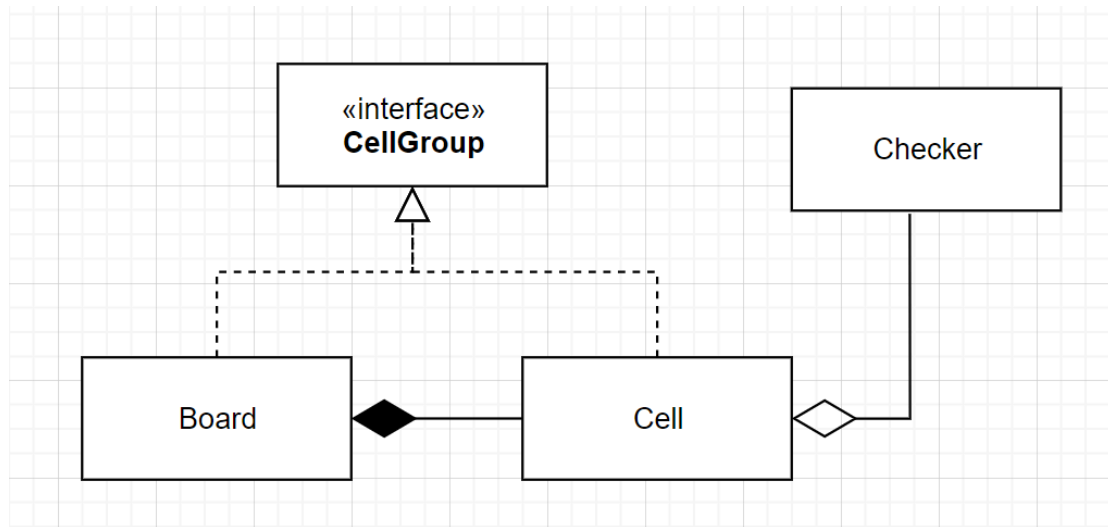
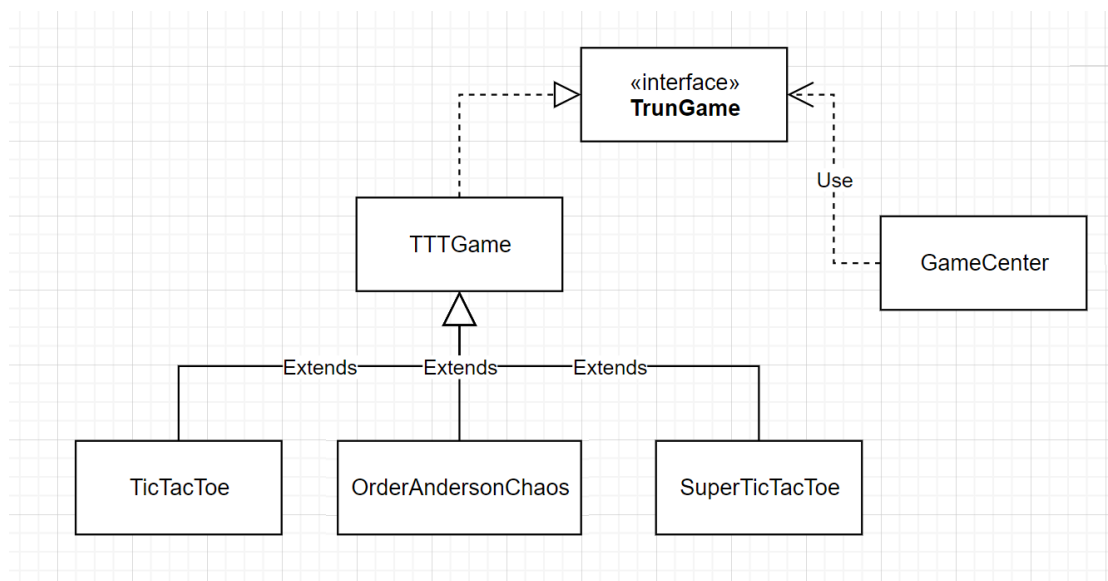


## Class structure



The class **Board** and **Cell** share an interface--**CellGroup** to form a composition design pattern. One board can contain another board as a cell. You can make the board include child boards, which makes it easier to implement Super Tic Tac Toe games. The code of the base class **TTTGame** is well reused.



`TurnGame` is the interface of turn-based games. Class `TTTGame` defines the game like Tic Tac Toe. Other game classes **TicTacToe**, **OrderAndChaos** and **SuperTicTacToe** inherit it. They simply override the necessary functional functions for customization. Class **gameCenter** can use other games which implement the same interface--**TurnGame**.

Because of the flexible design of the board, it can handle rectangular chessboards and customize the number of linkages. Therefore Class `TTTGame` as base for 3 other game classes will reduce code repetition.

## **Adding turn based variants**

If the another "turn based" variant game is close to TTT, you can directly inherit TTT and customize your own features. If the new game and TTT are very different, you only need to implement the interface `TurnGame`, and then implement your own function. Because the board's design is flexible, new games can be used directly. The logic of `GameCenter` does not need to be changed.