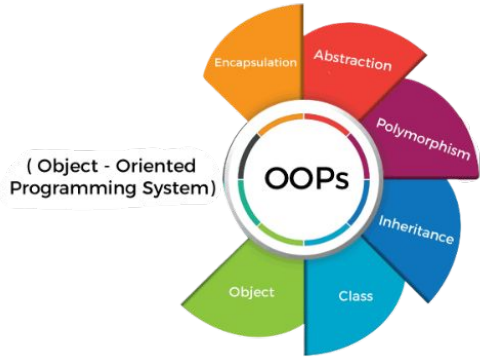
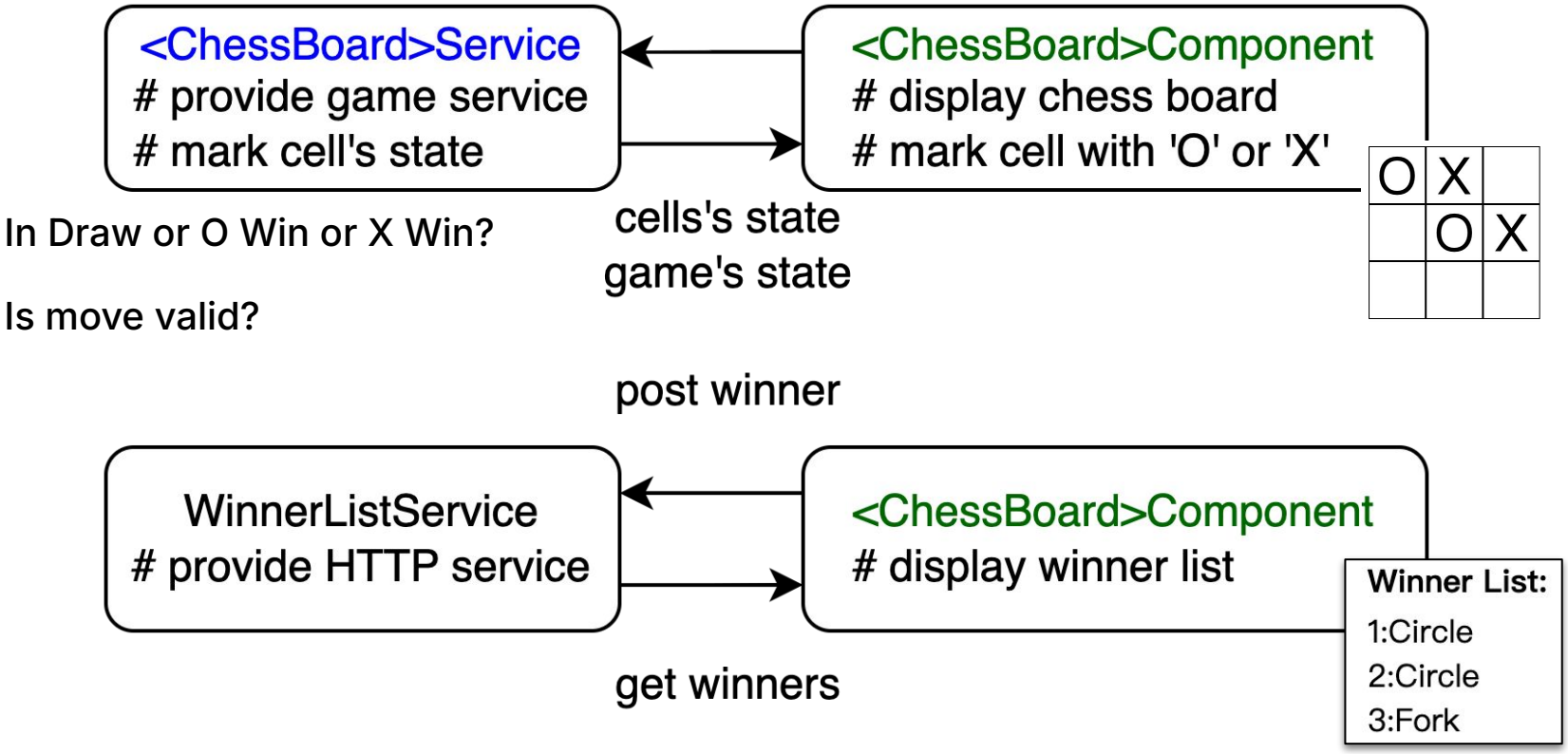
A Chess Game Website

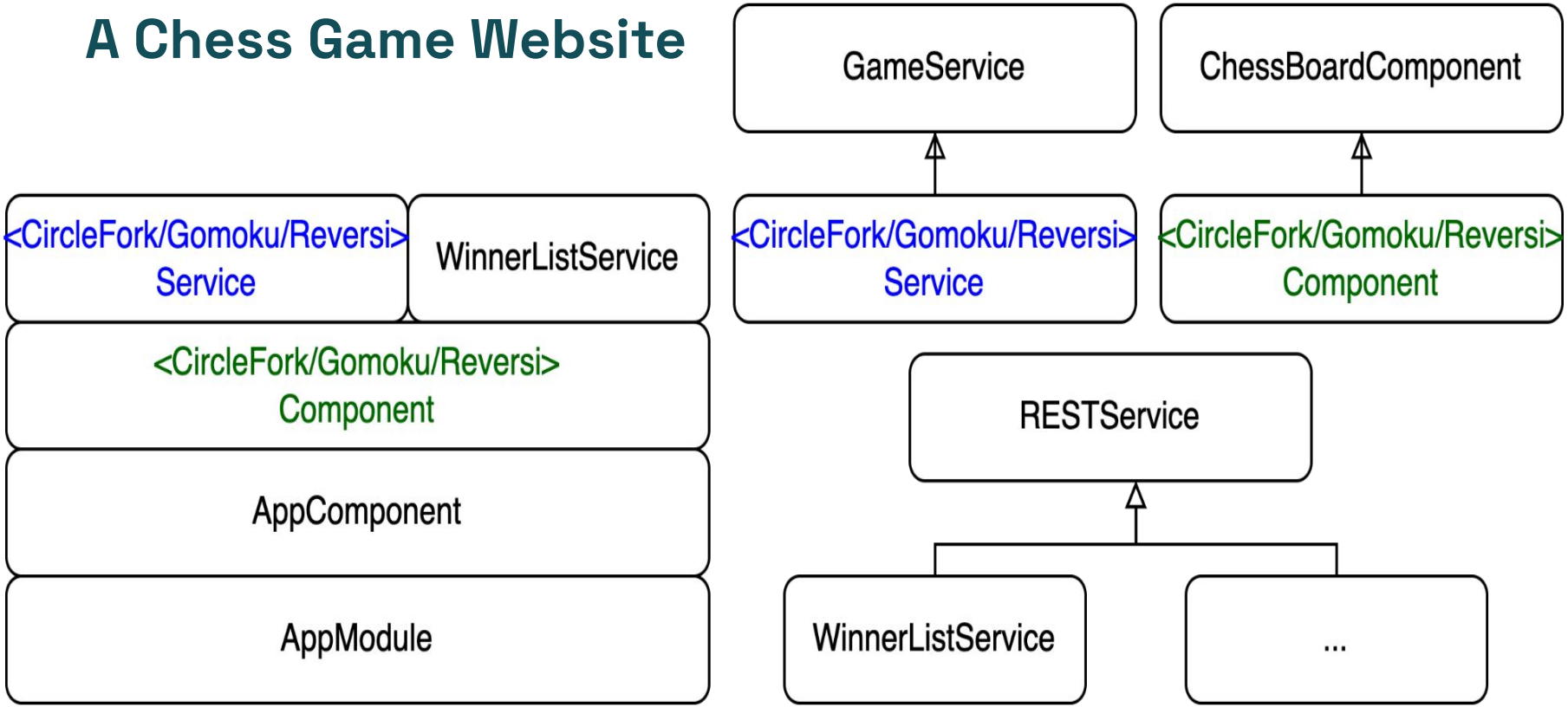


display: https://stackblitz.com/github.com/JohnBlue-git/ChessBoardGameWeb

A Chess Game Website



A Chess Game Website



A Chess Game Website

Developing flow:

Ask ChatGpt can you create a circle fork game in angular

↓

Scan over the results (variables, components, functions)
and ponder on:
what should be the functions of circle fork game's component?
what should be the functions of circle fork game's game service?
what are the similarity among circle-fork, gomoku, and reversi?
Can any common parts be grouped into the base class?

↓

Revised the code structure according to the contemplation

↓

Ask ChatGpt can you create a gomoku or reversi game,
and seek to integrate the code provided into our previous solution.

Skills used

- **Angular Framwork:** folder's struture / package and application settings / import modules / html and css layout / etc ...
- Unit Test: **Karma and Jasmine** (Testing environment and Assertion library).
- Apply **OOP design** to craft 3 types of chess board games and their corresponding game logic service, and integrate the components with the services using **Dependency Injection**.
- Properly use interface, abstract, class and **boost code reusability**
- Use **BrowserModule** to set routes.
- Use **HttpClientModule** to perform HTTP post, get, ...
- Adopt **DTO pattern** to **prevent over-posting attck**.

SOID principle applied

- **Single-responsibility Principle:**

  Divide the functions component class and service class to make each reseponsibility clear and simple.In addition, it also make unit test become more easy.

- **Open-closed Principle:**

  ChessBoardComponent(base class) is closed to change but it is open to derived into different <ChessBoard>Components.

- **Dependency Inversion Principle:**

  HTTP or game services are all defined as abstract interface in the higher module, and those lower modules will be inject via Dependency Injection mechanism.

Difficulties encountered

Angular:

- The toturials on internet could often be outdated. It takes time to research and figure out the newest standard.
- Resolve package dependencies and identify appropriate package settings and versions could be tedious. Also, for unit test, additional pakcges should be included.
- Complete the unit tests. The situations in game are complicated, especially when the chess board is wide and ther are too many cells to be considered.

What I have noticed about using ChaptGpt

Opportunities:

- AI can quickly generate basic HTML, CSS, and target components when given exact orders.
- It is nice to ask AI to assist with game logic, such as whether a move is valid or who wins. (such as Reversi)
- Another benefit is the naming conventions. Their proposed naming is helpful.

Attentions:

- Please keep in mind that the code may not be suitable for direct execution due to minor incorrect in logic or poor design pattern. Thus, sufficient knowledge regard the code syntax, the target logic intending to apply, and what will be a better design pattern are still required.

How I ask to ChatGPT

- In angular, can you give me a component named circle-fork for playing circles and forks game, which has an interface with a function to mark circle or fork, a function to check who wins, a function to reset chess board.

- Can you write unit test for the following angular component:

  <...code...>

- In angular, can you implement how to check the movement is valid in Reversi game, considering chess board start from top to bottom and from left to right.