

zustand

<https://zustand.docs.pmnd.rs/guides/tutorial-tic-tac-toe>

```
zustand-tic-tac-toe/  
├─ public/           # 정적 파일  
│  
├─ src/              # React 소스 파일  
│   ├─ components/  # 컴포넌트 폴더  
│   │   ├─ Board.jsx  
│   │   └─ Square.jsx  
│   │  
│   └─ store/        # Zustand 상태관리 폴더  
│       └─ gameState.js  
│  
├─ App.jsx           # 메인 컴포넌트  
├─ main.jsx          # 진입 파일  
│  
├─ index.html        # HTML 템플릿  
└─ package.json      # 의존성 및 스크립트
```

Zustand 상태 관리는 `gameStore.js` 에 작성

컴포넌트는 `components/` 폴더에 각각 작성

구조적으로 깔끔하고 상태 관리와 UI가 분리된 형태

Zustand의 사용법 요약

1. `create` 함수를 사용해 상태를 정의한다.
2. 상태는 훅 형태로 `useGameStore` 에서 제공된다.
3. `set` 을 통해 상태를 변경하며, 상태 변경 시 컴포넌트가 자동으로 리렌더링된다.
4. 상태는 전역으로 공유되며, 여러 컴포넌트에서 동일한 상태를 사용 가능하다.

1. Zustand 상태 관리 코드 (`src/store/gameStore.js`)

Zustand를 사용하여 상태를 관리하는 코드

상태는 `useGameState` 라는 커스텀 훅을 통해 관리

주요 부분 설명

- `create` : Zustand 상태를 정의할 때 사용하는 함수
- 상태는 객체 형태로 정의되고, 그 안에 `board`, `isXNext`, `winner` 와 같은 값들이 있다.
- `set` : 상태를 변경할 때 사용하는 함수
ex)
`playMove` 함수에서 게임 보드 상태를 변경하고 승자를 계산한 후에 다시 상태를 반환
- `useGameState` 는 상태를 관리하는 훅으로, 컴포넌트에서 이 훅을 호출하면 상태를 사용할 수 있다.

2. Square 컴포넌트 (`src/components/Square.jsx`)

틱택토의 각 칸을 나타내는 `Square` 컴포넌트

`Square` : 사용자가 클릭할 수 있는 버튼을 렌더링

주요 부분 설명

- `Square` : 게임 보드의 각 칸을 표현
- `value` : 해당 칸에 X 또는 O를 표시하는 값
- `onClick` : 클릭 이벤트를 처리하는 함수로, `playMove` 가 호출된다.

3. Board 컴포넌트 (`src/components/Board.jsx`)

`Board` : 게임 보드 전체를 담당하는 컴포넌트
9개의

`Square` 컴포넌트를 그리드 형태로 표시

주요 부분 설명

- `board` 배열을 사용하여 9개의 `Square` 컴포넌트를 그리드 형태로 표시한다.
 - 각 칸에 `value` 를 넘겨주고, 클릭 시 `playMove` 함수를 호출하여 게임을 진행한다.
 - 게임의 승자가 결정되면 `winner` 값을 표시한다.
 - 게임을 재시작할 수 있는 `Restart` 버튼을 제공한다.
-
- `Board` 는 `board` 배열을 사용해 9개의 `Square` 컴포넌트를 그린다.
 - 각 칸을 클릭하면 `playMove` 함수가 호출되어 게임 상태가 업데이트된다.
 - 게임이 끝났다면 승자(`winner`)를 표시하고, "Restart" 버튼을 통해 게임을 초기화할 수 있다.

4. App 컴포넌트 (`src/App.jsx`)

`Board` 컴포넌트를 불러와서 화면에 표시하는 최상위 컴포넌트

5. main.jsx

React 애플리케이션의 진입점, `App` 컴포넌트를 루트 DOM에 렌더링