

Mediator

دیزاین پترن هادر React

Behavioral Patterns

به زبان ساده همراه با مثال های عملی
کاربرد دیزاین پترن ها برای برنامه نویسان React



Behavioral Patterns

Mediator



الگوی Mediator ارتباط بین اجزای مختلف یک سیستم را از طریق یک شی واسطه مدیریت می کند. در این الگو به جای اینکه اجزای سیستم بصورت مستقیم با هم ارتباط برقرار کنند، از طریق یک شی واسطه با هم در ارتباط خواهند بود.

فرض کنید می خواهیم با توجه به سناریو زیر یک بازی برای تاس انداختن بسازیم.

در این بازی سه بازیکن داریم و بازیکنی که تاس را می اندازد خودش نباید نتیجه را بیند اما سایر بازیکنان نتیجه او را دریافت می کند.

بخش Interface

در این بخش یک Interface به نام Colleague می سازیم که برای مشخص نمودن بازیکنان است و دارای یک فیلد id برای منحصر به فرد نمودن بازیکنان و یک Method به نام receiveMessage برای دریافت پیام می باشد.



```
● ● ●  
export default interface Colleague {  
  id: string;  
  receiveMessage(message: string): void;  
}
```

بخش Interface

در این بخش یک Interface می سازیم که وظیفه ارسال پیام به بازیکنان را بر عهده دارد.



Mediator

```
import Colleague from "./colleague.interface";

export default interface Mediator {
  sendMessage(message: string, sender: Colleague): void;
}
```

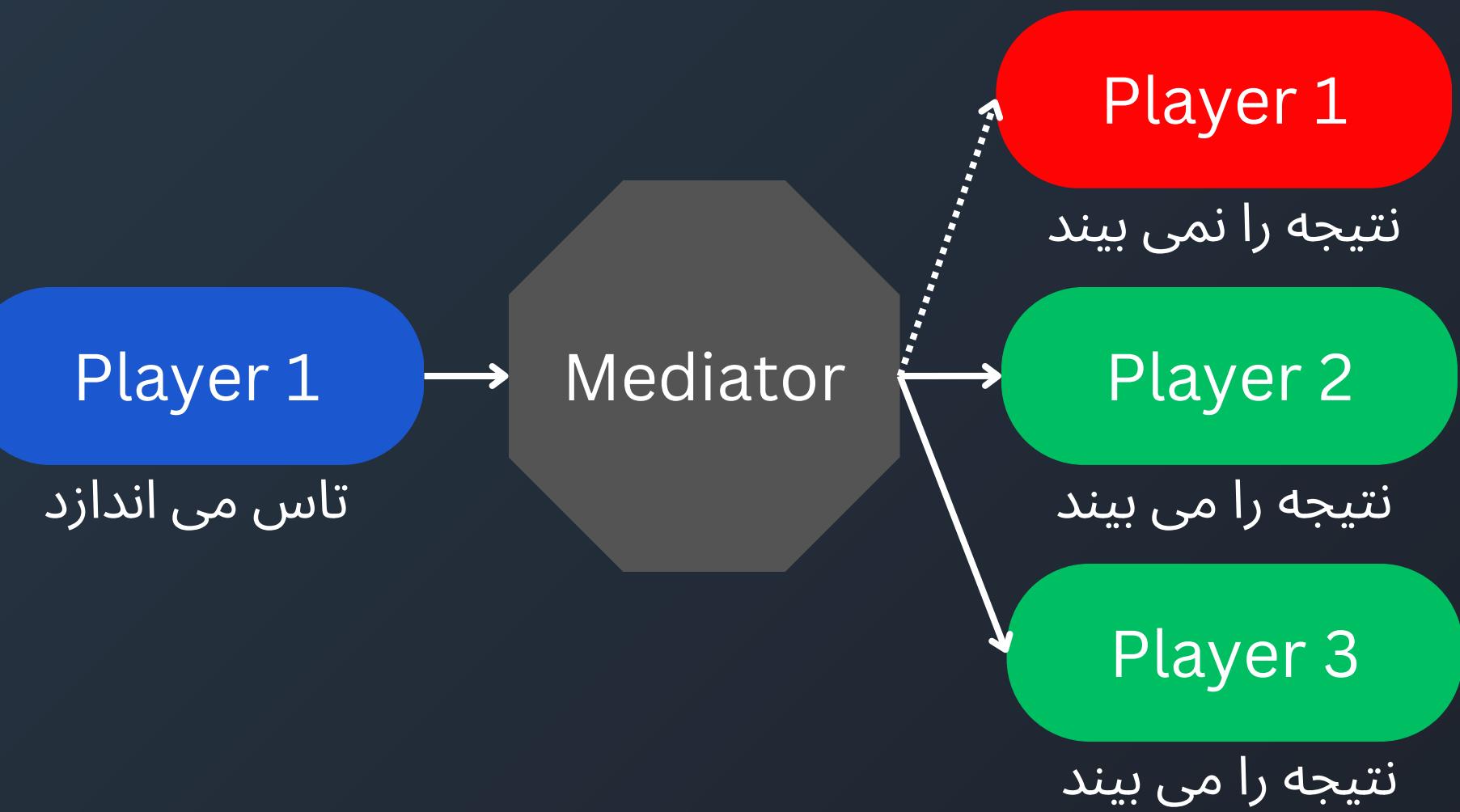
بخش ConcreteMediator

در این بخش Mediator را پیاده سازی می کنیم که دارای دو Method برای ثبت نام بازیکن و ارسال پیام به بازیکنان می باشد.



ConcreteMediator

```
●●●  
import Colleague from './interfaces/colleague.interface';  
import Mediator from './interfaces/mediator.interface';  
  
export default class ConcreteMediator implements Mediator {  
    private colleagues: Colleague[] = [];  
  
    public register(colleague: Colleague): void {  
        const exists = this.colleagues.some(x => x.id === colleague.id);  
        if (!exists) {  
            this.colleagues.push(colleague);  
        }  
    }  
  
    public sendMessage(message: string, sender: Colleague): void {  
        this.colleagues.forEach(colleague => {  
            if (colleague.id !== sender.id) {  
                colleague.receiveMessage(message);  
            }  
        });  
    }  
}
```



بخش PlayerProps

در این بخش Props مربوط به Player را به شکل زیر می سازیم.



PlayerProps

```
● ● ●  
import ConcreteMediator from '../concrete-mediator';  
  
export default interface PlayerProps {  
    name: string;  
    mediator: ConcreteMediator;  
}
```

بخش Style برای Player

به جهت ظاهر زیبا در لحظه دریافت پیام توسط بازیکنان این بخش را اضافه می کنیم.



```
● ● ●  
.content {  
    font-size: 12px;  
    font-weight: 700;  
    padding: 5px 10px;  
    margin: 5px;  
    border-radius: 5px;  
    transition: all 1s ease;  
}  
  
.content.highlight {  
    background-color: lime;  
    box-shadow: 0 0 10px lime;  
    color: black;  
}
```

بخش Player

در این بخش Player Component مربوط به شکل زیر می سازیم.



Player

```
import { useEffect, useMemo, useState } from 'react';
import { v4 as uuid } from 'uuid';
import Colleague from './interfaces/colleague.interface';
import PlayerProps from './player.props';
import './style.css';

const Player: React.FC<PlayerProps> = ({ name, mediator }) => {
  const [message, setMessage] = useState<string>('');
  const [highlight, setHighlight] = useState(false);

  const player: Colleague = useMemo(
    () => ({
      id: uuid(),
      receiveMessage: (message: string) => {
        setMessage(message);
        highlighter();
      },
      []
    }),
    []
  );

  useEffect(() => {
    mediator.register(player);
  }, [mediator, player]);

  const handleClick = () => {
    const roll = Math.floor(Math.random() * 6) + 1;
    mediator.sendMessage(`${name} rolled a ${roll}`, player);
  };

  const highlighter = () => {
    setHighlight(true);
    setTimeout(() => setHighlight(false), 1000);
  };

  return (
    <div>
      <h3>{name}</h3>
      <button onClick={handleClick}>Roll Dice</button>
      {message && (
        <span className={`content ${highlight ? 'highlight' : ''}`}>{message}</span>
      )}
    </div>
  );
};

export default Player;
```



بخش Game

در این بخش برای Game و تعریف بازیکنان یک Component به شکل زیر می سازیم.



```
● ● ●

import ConcreteMediator from './pattern/concrete-mediator';
import Player from './pattern/Player/Player';

const Game: React.FC = () => {
  const mediator = new ConcreteMediator();

  return (
    <div>
      <Player mediator={mediator} name='Player 1' />
      <hr />
      <Player mediator={mediator} name='Player 2' />
      <hr />
      <Player mediator={mediator} name='Player 3' />
    </div>
  );
};

export default Game;
```

طرز استفاده از Game

در پایان به شکل زیر از Game استفاده می کنیم.



MediatorDesignPattern



```
import Game from './Game';

const MediatorDesignPattern = () => {
  return (
    <div>
      <Game />
    </div>
  );
};

export default MediatorDesignPattern;
```