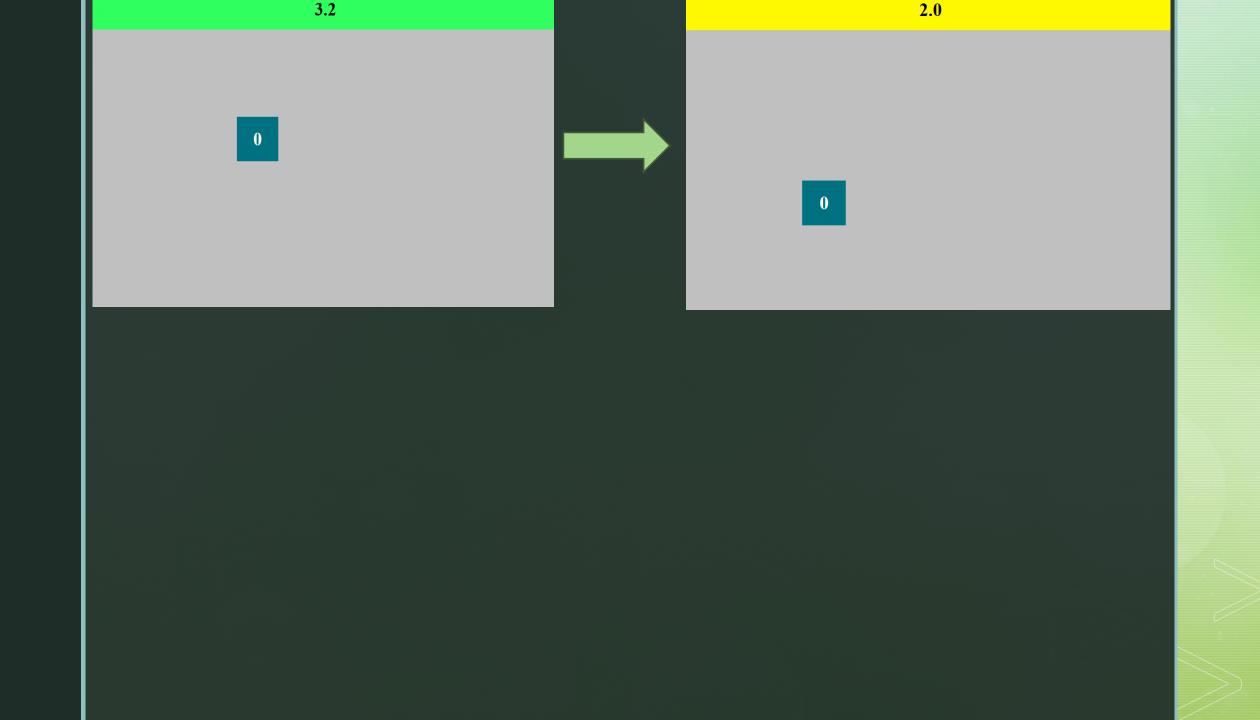
OVRT Házi Kovács Marcell BCGDH7

Tervezési minták:

- ObserverMiddleware

Observer minta célja:

Azt a célt szolgálja, hogy a timer, illetve a score attribútumait módosítsa (szín, stb)



Middleware minta célja:

Ez a minta szolgál a kattintás hatására történő dolgok elvégzéséért

```
You, 31 minutes ago | 1 author (You)
24 vexport class cubeClickMiddleware implements Middleware {
      process(context: any): any {
         if (context.type === 'cubeClick') { // set the cube's color when clicked, then reset
           const cube = document.getElementById("cube")
          if (cube) {
            cube.style.background = "#0091a4";
            setTimeout(() => {
              cube.style.backgroundColor = "#007180";
             }, 100);
         if (context.type === 'cubeClick' && context.speed <= 0.4) { // increase the cube's speed until 0,4
           context.speed += 0.005;
         if (context.type === 'cubeClick' && context.changeHorizontalChance <= 0.01) { // increase the chance of direction change until 0.01
38 🗸
           context.changeHorizontalChance += 0.00005
           context.changeVerticalChance += 0.00005
                                                                    clickedOnCube() {
                                                                      // setting context to the middleware
                                                            92
                                                                      const context = {
44
                                                                         type: 'cubeClick',
                                                                         speed: this speed.
                                                            95
                                                                         timeLeft: this.gameState.timeLeft,
                                                                         changeHorizontalChance: this.changeHorizontalChance,
                                                                         changeVerticalChance: this.changeVerticalChance,
                                                            98
                                                                      this.middlewareService.execute(context);
                                                                      this.speed = context.speed;
                                                           102
                                                                      this.gameState.timeLeft += 0.3;
                                                                      this.gameState.score += 1;
                                                           105
```

MainComponent- gameState: GameState - speed: number - directionRight: boolean - directionDown: boolean - showStartButton: boolean - changeVerticalChance: number - changeHorizontalChance: number - middlewareService: MiddlewareService + ngOnInit(): Promise<void> + startGame(): void + handleCubeMovement(): Promise<void> + timer(): Promise < void> + delay(ms: number): Promise < void> + clickedOnCube(): void + reset(): void + updateScoreDisplay(newScore: number): void + updateTimerDisplay(newTimeLeft: number): void uses GameState - scoreSubject: Subject<number> - timeLeftSubject: Subject<number> - score: number - timeLeft: number + score: number + timeLeft: number + subscribeToScore(observer: (score: number) => void): void + subscribeToTimeLeft(observer: (timeLeft: number) => void): void has-a Subject<T> - observers: Array<(state: T) => void> + subscribe(observer: (state: T) => void): void + unsubscribe(observer: (state: T) => void): void

+ notify(state: T): void

Middleware Service - middlewares: Array < Middleware > + use(middleware: Middleware): void + execute(context: any): void uses Middleware + execute(context: any, next: () => void): void