

Benaia J. Heinze Silva

Dilluns 10:30

BattleShip development with TDD

### *Funcionalitat: Registrar impactes al vaixell*

- **Descripció:** Incrementa el comptador d'impactes rebuts pel vaixell.
- **Localització:**
  - Arxiu: `Ship.java`
  - Classe: `Ship`
  - Mètode: `hit()`
- **Test:**
  - Arxiu: `ShipTest.java`
  - Classe: `ShipTest`
  - Mètode: `testHit()`
  - **Descripció del tipus de test:**
    - **Tècnica:** Mockups per simular impactes successius fins a enfonsar el vaixell.

### *Funcionalitat: Definir ubicacions de vaixells*

- **Descripció:** Crea una ubicació que defineix posició, mida i orientació del vaixell.
- **Localització:**
  - Arxiu: `ShipPlacement.java`
  - Classe: `ShipPlacement`
  - Mètode: Constructor i validacions associades.
- **Test:**
  - Arxiu: `ShipPlacementTest.java`
  - Classe: `ShipPlacementTest`
  - Mètode: `testValidPlacement()` i `testInvalidPlacement()`
  - **Descripció del tipus de test:**
    - **Tècnica:** Caixa negra per validar coordenades vàlides i invàlides.

### **Funcionalitat: Col·locació de vaixells al tauler**

- **Descripció:** Col·loca un vaixell en una posició especificada del tauler.
- **Localització:**
  - Arxiu: `Board.java`
  - Classe: `Board`
  - Mètode: `placeShip(ShipPlacement placement)`
- **Test:**
  - Arxiu: `BoardTest.java`
  - Classe: `BoardTest`
  - Mètode: `testPlaceShip()`
  - **Descripció del tipus de test:**
    - **Tècnica:** Caixa blanca amb cobertura de sentències per validar col·locacions vàlides i no vàlides.

### *Funcionalitat: Registrar un dispar*

- **Descripció:** Marca un dispar a les coordenades especificades i retorna si el dispar ha estat encertat.
- **Localització:**
  - Arxiu: `Board.java`
  - Classe: `Board`
  - Mètode: `shoot(int x, int y)`
- **Test:**
  - Arxiu: `BoardTest.java`
  - Classe: `BoardTest`
  - Mètode: `testShoot()`
  - **Descripció del tipus de test:**
    - **Tècnica:** Particions equivalents per validar dispars vàlids, repetits i fora de límits.

### *Funcionalitat: Verificar si tots els vaixells estan enfonsats*

- **Descripció:** Comprova si tots els vaixells del tauler han estat enfonsats.
- **Localització:**
  - Arxiu: `Board.java`
  - Classe: `Board`
  - Mètode: `allShipsSunk()`
- **Test:**
  - Arxiu: `BoardTest.java`
  - Classe: `BoardTest`
  - Mètode: `testAllShipsSunk()`
  - **Descripció del tipus de test:**
    - **Tècnica:** Mockups per simular diferents estats del tauler.
    - **Cobertura de camins:** Cobrir escenaris amb tots els vaixells intactes, parcialment enfonsats i completament enfonsats.

### **Funcionalitat: Generar moviments de la IA**

- **Descripció:** Genera automàticament un moviment vàlid per a un jugador controlat per IA.
- **Localització:**
  - Arxiu: `Player.java`
  - Classe: `Player`
  - Mètode: `makeMove()`
- **Test:**
  - Arxiu: `PlayerTest.java`
  - Classe: `PlayerTest`
  - Mètode: `testMakeMove()`
  - **Descripció del tipus de test:**
    - **Tècnica:** Caixa negra per validar que els moviments generats són dins del tauler i no repetitius.
    - **Mockups:** Simular un tauler per validar coordenades generades.

### **Funcionalitat: Generació de moviments automàtics de la IA**

- **Descripció:** Permet a un controlador d'IA generar moviments automàtics en el tauler, evitant cel·les ocupades o disparades.
- **Localització:**
  - Arxiu: `AIController.java`
  - Classe: `AIController`
  - Mètode: `makeAIMove(char[][] grid)`
- **Test:**
  - Arxiu: `BattleShipControllerTest.java`
  - Classe: `BattleShipControllerTest`
  - Mètode: `Suggereix testMakeAIMove()`
  - **Descripció del tipus de test:**
    - **Tècnica:** Caixa negra basada en particions equivalents per validar coordenades vàlides.

### **Funcionalitat: Control del flux del joc**

- **Descripció:** Gestiona tot el procés del joc, des de la inicialització fins al final, incloent col·locacions i torns.
- **Localització:**
  - Arxiu: BattleshipController.java
  - Classe: BattleshipController
  - Mètode: startGame()
- **Test:**
  - Arxiu: BattleShipControllerTest.java
  - Classe: BattleShipControllerTest
  - Mètode: testStartGame()
  - **Descripció del tipus de test:**
    - **Tècnica:** Caixa blanca amb cobertura de sentències i branques.
    - **Mockups:** S'utilitzen per simular la vista i el comportament dels jugadors.

### **Funcionalitat: Col·locació de vaixells pel jugador**

- **Descripció:** Permet al jugador col·locar els seus vaixells manualment al tauler.
- **Localització:**
  - Arxiu: BattleshipController.java
  - Classe: BattleshipController
  - Mètode: placePlayerShips(Player player)
- **Test:**
  - Arxiu: BattleShipControllerTest.java
  - Classe: BattleShipControllerTest
  - Mètode: testPlacePlayerShips()
  - **Descripció del tipus de test:**
    - **Tècnica:** Mockups per simular la interacció de l'usuari amb la vista.
    - **Particions equivalents:** Validació d'ubicacions vàlides i no vàlides.

### **Funcionalitat: Torn de joc**

- **Descripció:** Executa un torn on l'atacant fa un moviment contra el defensor.
- **Localització:**
  - Arxiu: BattleshipController.java
  - Classe: BattleshipController
  - Mètode: playTurn(Player attacker, Player defender)
- **Test:**
  - Arxiu: BattleShipControllerTest.java
  - Classe: BattleShipControllerTest
  - Mètode: testPlayTurn()
  - **Descripció del tipus de test:**
    - **Tècnica:** Mockups per simular interacció i cobrir casos de dispar encertat o fallat.

### ***Funcionalitat: Mostrar l'estat del joc a l'usuari***

- **Descripció:** Presenta missatges i l'estat del tauler al jugador durant el joc.
- **Localització:**
  - Arxiu: BattleshipView.java
  - Classe: BattleshipView
  - Mètodes: `displayWelcomeMessage()`, `displayGameOver()`, `displayShotResult(boolean hit)`.