

Report Practiques

Game:

Funcionalitat:	El main crea un Game que aquest crea una classe Menu().
Localització:	Develop : Game funció main Test: GameTest i funció TestGame .
Test:	Caixa blanca: Statement <u>Coverage</u> . Per tal de comprovar que al iniciar el Game es crea un Menu y es mostren les diferents opcions.

Menú:

Totes les classes utilitzen **ManagerIOMock()** per simular les entrada de l'usuari per teclat:

Funcionalitat:	Mostra el menú con las diferents opcions y executa la Opció 1 Jugar, que comença la partida.
Localització:	Develop : Menu class Test: MenuTest I function MenuTestRunning
Test:	Caixa blanca: Condition Coverage/ Statement Coverage/Path Coverage . Per tal de comprovar que el passem per totes les condicions del Switch, en aquest cas l'opció 1 de començar a Jugar. Comprovem comparant amb el Sting que mostra al començar a jugar.

Funcionalitat:	Mostra el menú con las diferents opcions y executa la Opció 2 Opcions, que mostra les diferents opcions.
Localització:	Develop : Menu class Test: MenuTest I function MenuTestOptions .
Test:	Caixa blanca: Condition Coverage/ Statement Coverage/Path Coverage . Per tal de comprovar que el passem per totes les condicions del Switch , en aquest cas l'opció 2 de mostrar les opcions. Comprovem comparant amb el String que mostra al seleccionar les opcions.

Funcionalitat:	Mostra el menú con las diferents opcions y executa la Opció 3 Exit, que tanca el Joc.
Localització:	Develop : Menu class Test: MenuTest I function MenuTestExit.
Test:	Caixa blanca: Condition Coverage / Statement Coverage /Path Coverage. Per tal de comprovar que el passem per totes les condicions del Switch, en aquest cas l'opció 2 de mostrar les opcions. Comprovem comparant amb el String que mostra al seleccionar les opcions.

Funcionalitat:	Comprovació de tots el casos que no son correctes, valors límit i frontera
Localització:	Develop : Menu class Test: MenuTest I function TestMenuBadOptions.
Test:	Caixa negra: Particions equivalent i valores límit/frontera.

En aquest cas concret al realitzar **el Condition Coverage** per a les diferents condicions del switch també estem realitzant **Statement coverage , Decision Coverage i Path Coverage.**

Jugador:

Totes les classes utilitzen **ManagerIOMock()** per simular les entrada de l'usuari per teclat:

Funcionalitat:	Creació de dos jugadors per verificar els diferents tipo (Jugador: 1 i Random: 2)
Localització:	Develop : Jugaro class i funcion Jugador() Contrustor Test: JugadorTest I funcion TestCreateJugador.
Test:	Caixa blanca: Statement Coverage / Path Coverage. Comprovem la correcta inicialització del contrustor.

Funcionalitat:	Fem que els dos jugadors utilitzen la funció d'atacar
Localització:	Develop : Jugaro class i i funcion atacar() Test: JugadorTest I funcion TestJugadorAtacar.
Test:	Caixa blanca: Condition Coverage / Statement Coverage /Path Coverage. Per tal de comprovar el dos path possibles fem que ataqüi el jugador i també el Random.

Funcionalitat:	Funció que indica si ja hi ha un guanyador en la partida
Localització:	Develop : Jugaro class i funcion ganador() Test: JugadorTest I funcion TestJugadorGana.
Test:	Caixa blanca: Condition Coverage / Statement Coverage /Path Coverage. Simulen una partida sencera per tal de comprovar el dos casos, quant encara no ha guanyat ningun i quant finalitza (guanya algu)

Partida:

Totes les classes utilitzen **ManagerIOMock()** per simular les entrada de l'usuari per teclat:

Funcionalitat:	Simulació d'una partida en què guanya el jugador
Localització:	Develop : Partida class i funcion printGanador() Test: PartidaTest I funcion TestGanarJugador.
Test:	Caixa blanca: Condition Coverage Simulen una partida sencera per tal de comprovar el dos casos, quant encara no ha guanyat ningun i quant finalitza (guanya algú).

Funcionalitat:	Simulació d'una partida en què guanya el random
Localització:	Develop : Partida class i funcion printGanador() Test: PartidaTest I funcion TestGanarRandom.
Test:	Caixa blanca: Condition Coverage Simulen una partida sencera per tal de comprovar el dos casos, quant encara no ha guanyat ningun i quant finalitza (guanya algú).

Simulant el dos casos realitzem també **Statement Coverage** i **Path Coverage**

<Cristian Vega Sanchez # 1426805>

<HundiLaFlota Dimarts 12:30-14:30>

Posicion:

Funcionalitat:	Creacio de posicions amb valors = X, Y , Tamaño , Orientacion
Localització:	Develop : Posicion class Test: PosicionTest I function TestConstructor.
Test:	Caixa blanca: Statement Coverage. Comprovació de los mètodes Constructor ,Setter, Getter i toString.

Funcionalitat:	Retornar la orientació en format String
Localització:	Develop : Posicion class Test: PartidaTest I function TestOrientacion.
Test:	Caixa blanca: Condition Coverage /Statement Coverage . Realitzem les diferents condicions de la orientació

Simulant el dos casos realitzem també Condition Coverage, Statement Coverage i Path Coverage de la classe Posicion.

Tablero:

Funcionalitat:	Mostrar el Tauler de la Partida
Localització:	Develop : Tablero class i funcion mostrarTablero() Test: TableroTest I funcion testMostrarTablero() .
Test:	Caixa blanca: Statement Coverage /Path Coverage. Assegura que es mostra correctament el Tauler().

Funcionalitat:	Inicialitzar tauler
Localització:	Develop : Tablero class i funcion iniciarTablero() Test: TableroTest I funcion testIniciarTablero()
Test:	Caixa blanca: Loop Testing / Statement Coverage /Path Coverage. Simulen una partida sencera per tal de comprovar el dos casos, quant encara no ha guanyat ningun i quant finalitza (guanya algú).

Funcionalitat:	Inserta una posició en el Tablero
Localització:	Develop : Tablero class i funcion inserirPosicion() Test: TableroTest I funcion testInserirPosicion ()
Test:	Caixa blanca: Path Coverage. Insertem una posició correcta.

Funcionalitat:	Inserta una posició incorrecta en el Tablero
Localització:	Develop : Tablero class i funcion inserirPosicionKO() Test: TableroTest I funcion testInserirPosicionKO ()
Test:	Caixa blanca: Path Coverage. Insertem una posició incorrecta.

inserirPosicion() + inserirPosicionKO() = Decision / Condition Coverage i Statement

Funcionalitat:	Inserta una posició genera de forma aleatòria
Localització:	Develop : Tablero class i funcion inserirPosicionRandom() Test: TableroTest I funcion testInserirPosicionRandom ()
Test:	Caixa blanca: Statement/ Path Coverage. Generem una posició correcta.

<Cristian Vega Sanchez # 1426805>

<HundiLaFlota Dimarts 12:30-14:30>

Funcionalitat:	Genera una barrera protectora al voltant d'una posició (1 quadrat)
Localització:	Develop : <code>Tablero</code> class i funcion <code>proteccionBarco()</code> Test: <code>TableroTest</code> I funcion <code>testProteccionBarco ()</code>
Test:	Caixa blanca: Condition Coverage / Statement Coverage /Path Coverage. Caixa Negra: Limites i Frontera. Orientació : (-inf, 0), [0, 3], [4, inf) Simulen una partida sencera per tal de comprovar el dos casos, quant encara no ha guanyat ningun i quant finalitza (guanya algú)

Funcionalitat:	Inserta un vaixell dintre del taulell.
Localització:	Develop : <code>Tablero</code> class i funcion <code>insertarBarco()</code> Test: <code>TableroTest</code> I funcion <code>testInsertarBarco ()</code>
Test:	Caixa blanca: Condition Coverage / Statement Coverage /Path Coverage. Caixa Negra: Limites i Frontera. Orientació : (-inf, 0), [0, 3], [4, inf) Inserta Barcos en todas las orientacions possibles.

Funcionalitat:	Comprova si es pot inserta una posició amb una orientacions concreta.
Localització:	Develop : <code>Tablero</code> class i funcion <code>sePuedeInsertarOrientacion()</code> Test: <code>TableroTest</code> I funcion <code>testSePuedeInsertarOrientacion ()</code>
Test:	Caixa blanca: Condition Coverage / Statement Coverage /Path Coverage. Caixa Negra: Limites i Frontera. Orientació : (-inf, 0), [0, 3], [4, inf) Inserta Barcos en todas las orientacions possibles.

Funcionalitat:	Agafa la orientació llegida per teclat
Localització:	Develop : <code>Tablero</code> class i funcion <code>leerOrientacion()</code> Test: <code>TableroTest</code> I funcion <code>testLeerOrientacion ()</code>
Test:	Caixa blanca: Condition Coverage / Statement Coverage /Path Coverage. Caixa Negra: Limites i Frontera. Orientació : (-inf, 0), [0, 3], [4, inf) <u>Insertem</u> Orientacions invalides i valides per cobrir tot el codi.

Funcionalitat:	Genera una orientació
Localització:	Develop : Tablero class i funcion generarOrientacion() Test: TableroTest I funcion testGenerarOrientacion ()
Test:	Caixa blanca: Condition Coverage / Statement Coverage /Path Coverage. Caixa Negra: Limites i Frontera. <u>Generem Orientacions invalides i valides per cobrir tot el codi.</u>

Funcionalitat:	Agafa la posició llegida per teclat (x-y)
Localització:	Develop : Tablero class i funcion leerXY() Test: TableroTest I funcion testLeerXY ()
Test:	Caixa blanca: Condition Coverage / Statement Coverage /Path Coverage. Caixa Negra: Limites i Frontera. Posicion x-y : (-inf, 1), [0, 10], [10, inf) <u>Lectura de posicions X,Y invalides i valides per cobrir tot el codi.</u>

Funcionalitat:	Genera una posició (x-y)
Localització:	Develop : Tablero class i funcion generarXY() Test: TableroTest I funcion testGenerarXY ()
Test:	Caixa blanca: Condition Coverage / Statement Coverage /Path Coverage. Caixa Negra: Limites i Frontera. Posicion x-y : (-inf, 1), [0, 10], [10, inf) <u>Generació de posicions X,Y invalides i valides per cobrir tot el codi.</u>

Funcionalitat:	El jugador atacar al tauler
Localització:	Develop : Tablero class i funcion atacar() Test: TableroTest I funcion testAtacar ()
Test:	Caixa blanca: Condition Coverage / Statement Coverage /Path Coverage. <u>Atacar a tots el tipus de posicions del tauler per cobrir tot el codi.</u>

Funcionalitat:	El jugador Aleatori atacar al tauler
Localització:	Develop : Tablero class i funcion atacarRandom() Test: TableroTest I funcion testAtacarRandom ()
Test:	Caixa blanca: Condition Coverage / Statement Coverage /Path Coverage. <u>Atacar a tots el tipus de posicions del tauler per cobrir tot el codi.</u>

<Cristian Vega Sanchez # 1426805>

<HundiLaFlota Dimarts 12:30-14:30>

JUnit

Finished after 0,124 seconds

Runs: 31/31 Errors: 0 Failures: 0

- PartidaTest [Runner: JUnit 5] (0,000 s)
 - testGanaRandom() (0,000 s)
 - testGanaJugador() (0,000 s)
- PosicionTest [Runner: JUnit 5] (0,000 s)
 - testOrientacion() (0,000 s)
 - testConstructor() (0,000 s)
- GameTest [Runner: JUnit 5] (0,001 s)
 - testGame (0,001 s)
- JugadorTest [Runner: JUnit 5] (0,004 s)
 - testJugadorGana (0,002 s)
 - testCreateJugador (0,001 s)
 - testJugadorAtacar (0,001 s)
- ManagerIOTest [Runner: JUnit 5] (0,004 s)
 - testManagerIOInInt (0,002 s)
 - testManagerIOGetRandomOrientacion (0,001 s)
 - testManagerIOOutString (0,001 s)
 - testManagerIOGetRandomInt (0,000 s)
 - testManagerIOOutInteger (0,000 s)
- MenuTest [Runner: JUnit 5] (0,003 s)
 - testMenuRunning (0,002 s)
 - testMenuOptions (0,000 s)
 - testMenuExit (0,001 s)
 - testMenuBadOption (0,000 s)
- TableroTest [Runner: JUnit 5] (0,004 s)
 - testLeerOrientacion (0,000 s)
 - testProteccionBarco (0,000 s)
 - testInsertarBarco (0,001 s)
 - testAtacarRandom (0,000 s)
 - testInsertPosicionKO (0,000 s)
 - testSePuedeInsertarOrientacion (0,001 s)
 - testInsertPosicion (0,000 s)
 - testGenerarXY (0,000 s)
 - testIniciarTablero (0,000 s)
 - testAtacar (0,001 s)
 - testGenerarOrientacion (0,000 s)
 - testLeerXY (0,000 s)
 - testInsertPosicionRandom (0,000 s)
 - testMostrarTablero (0,001 s)

▼ Battleship	98,6 %	5.027	73	5.100
> tests	97,2 %	2.516	73	2.589
▼ src	100,0 %	2.511	0	2.511
▼ com.battelship.main	100,0 %	1.458	0	1.458
> Game.java	100,0 %	15	0	15
> Jugador.java	100,0 %	166	0	166
> ManagerIO.java	100,0 %	49	0	49
> Menu.java	100,0 %	54	0	54
> Partida.java	100,0 %	97	0	97
> Posicion.java	100,0 %	78	0	78
> Tablero.java	100,0 %	999	0	999
▼ com.battelship.utils	100,0 %	1.053	0	1.053
> Constants.java	100,0 %	1.053	0	1.053

Game:

Funcionalitat: El main crea un Game que a su vez ejecuta el menú principal del juego
<descripció de la funcionalitat introduïda o modificada>

Localització: Game function main <Arxiu, classe i mètode desenvolupat>

Test: GameTest I function TestGame.

Caixa blanca: Statement Coverage. Per tal de comprovar que al iniciar el Game es crea un Menu y es mostren les diferents opcions.

<Descripció del tipus de test realitzat (caixa negra, caixa blanca, ...) i les tècniques utilitzades (particions equivalents, pairwise testing, mockups, automatització)>

SIEMPRE:

Caixa Negra:

- Particions Equivalents
- Valor límit i frontera

Caixa blanca:

- Statement Coverage
- Decision Coverage0 : minimo de 2 metodos
- Path Coverage
- Loop Testing (loop simple) n y m --

Caixa Negra	
Particions Equivalents	
Valor límit i frontera	

Caixa Blanca:	
Statement Coverage	OK
Decision Coverage	OK (minimo 2 metodos)
Path Coverage	OK
Loop Testing	OK