



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Facultat d'Informàtica de Barcelona



SCRABBLE

Projecte de Programació

Curs 2024-2025, QP

Versió 1.0

Grup 43.3

Escofet González, Gina (gina.escofet)

Gascón Moliné, Gerard (gerard.gascon)

Martínez Lassalle, Felipe (felipe.martinez.lassalle)

Pérez Silvestre, Biel (biel.perez)

Usero Martínez, Albert (albert.usero)

Índex

1. Decisions preses.....	11
1.1. Capa de dades.....	11
1.2. Capa de domini.....	11
1.3. Capa de presentació.....	12
1.4. Gestió de dependències.....	13
1.5. Testing.....	14
2. Casos d'ús.....	15
2.1. Diagrama de casos d'ús.....	15
2.2. Actors.....	15
2.2.1. Player.....	15
2.2.2. CPU.....	15
2.3. Descripció individual dels casos d'ús.....	16
2.3.1. SeeRanking.....	16
2.3.2. PauseGame.....	16
2.3.3. SetBoardSize.....	16
2.3.4. SetLanguage.....	16
2.3.5. SetNumPlayers.....	16
2.3.6. ThinkMove.....	16
2.3.7. MakeMove.....	17
2.3.8. PutPiece.....	17
2.3.9. DrawPiece.....	17
2.3.10. SkipTurn.....	17
2.3.11. SaveGame.....	17
2.3.12. LoadGame.....	17
3. Classes.....	18
3.1. Capa de dades ^[OBJ]	19
3.1.1. Diagrama.....	19
3.1.2. Descripció.....	20
3.1.2.1. Board.....	20
3.1.2.2. BoardType.....	20
3.1.2.3. JuniorBoard.....	20
3.1.2.4. PremiumTileType.....	21
3.1.2.5. StandardBoard.....	21
3.1.2.6. SuperBoard.....	21
3.1.2.7. CrossChecks.....	22
3.1.2.8. CatalanCrossChecks.....	22
3.1.2.9. EnglishCrossChecks.....	22
3.1.2.10. SpanishCrossChecks.....	22
3.1.2.11. DAWG.....	22
3.1.2.12. Node.....	23
3.1.2.13. PlayerDoesNotHavePieceException.....	23
3.1.2.14. ScrabbleException.....	23

3.1.2.15. Leaderboard.....	23
3.1.2.16. Score.....	23
3.1.2.17. Bag.....	23
3.1.2.18. Piece.....	24
3.1.2.19. GameProperties.....	24
3.1.2.20. Language.....	24
3.1.2.21. Movement.....	24
3.1.2.22. Player.....	25
3.2. Capa de domini.....	26
3.2.1. Diagrama.....	26
3.2.2. Descripció.....	27
3.2.2.1. DrawActionMaker.....	27
3.2.2.2. PlaceActionMaker.....	27
3.2.2.3. SkipActionMaker.....	27
3.2.2.4. AI.....	27
3.2.2.5. CatalanAI.....	27
3.2.2.6. EnglishAI.....	27
3.2.2.7. SpanishAI.....	27
3.2.2.8. IBoard.....	28
3.2.2.9. PointCalculator.....	28
3.2.2.10. PresentPiecesWordCompleter.....	28
3.2.2.11. WordGetter.....	28
3.2.2.12. WordPlacer.....	29
3.2.2.13. CrossCheckUpdater.....	29
3.2.2.14. WordAdder.....	30
3.2.2.15. WordValidator.....	30
3.2.2.16. InitialMoveNotInCenterException.....	30
3.2.2.17. MovementOutsideOfBoardException.....	30
3.2.2.18. NotEnoughPiecesInBagException.....	30
3.2.2.19. NotEnoughPiecesInHandException.....	30
3.2.2.20. WordDoesNotExistsException.....	31
3.2.2.21. WordNotConnectedToOtherWordsException.....	31
3.2.2.22. GameStepper.....	31
3.2.2.23. IWinScreen.....	31
3.2.2.24. GamePlayedLeaderboard.....	31
3.2.2.25. GamesWinsPair.....	31
3.2.2.26. GamesWonLeaderboard.....	31
3.2.2.27. MaxScoreLeaderboard.....	32
3.2.2.28. PlayerValuePair.....	32
3.2.2.29. TotalScoreLeaderboard.....	32
3.2.2.30. WinRateLeaderboard.....	32
3.2.2.31. MovementBoundsChecker.....	32
3.2.2.32. MovementCleaner.....	32
3.2.2.33. BagFiller.....	32

3.2.2.34. CatalanPiecesConverter.....	33
3.2.2.35. EnglishPiecesConverter.....	33
3.2.2.36. HandFiller.....	33
3.2.2.37. IFileReader.....	33
3.2.2.38. PieceDrawer.....	33
3.2.2.39. PieceGenerator.....	33
3.2.2.40. PiecesConverter.....	34
3.2.2.41. PiecesConverterFactory.....	34
3.2.2.42. PiecesInHandGetter.....	34
3.2.2.43. PiecesInHandVerifier.....	34
3.2.2.44. SpanishPiecesConverter.....	34
3.2.2.45. Endgame.....	34
3.2.2.46. IGamePlayer.....	35
3.2.2.47. Turn.....	35
3.2.2.48. TurnResult.....	35
3.2.2.49. AnchorUpdater.....	35
3.3. Capa de presentació.....	36
3.3.1. Diagrama.....	36
3.3.2. Descripció.....	37
3.3.2.1. DictionaryReader.....	37
3.3.2.2. LocaleReader.....	37
3.3.2.3. PiecesReader.....	37
3.3.2.4. SceneObjectWithParametrizedConstructorException.....	37
3.3.2.5. Scene.....	37
3.3.2.6. SceneManager.....	38
3.3.2.7. SceneObject.....	38
3.3.2.8. HandView.....	39
3.3.2.9. MovementParser.....	39
3.3.2.10. AIPlayerObject.....	39
3.3.2.11. HumanPlayerObject.....	39
3.3.2.12. PlayerObject.....	40
3.3.2.13. GameScene.....	40
3.3.2.14. MenuScene.....	41
3.3.2.15. Reader.....	41
3.3.2.16. BoardView.....	41
3.3.2.17. GameCreator.....	41
3.3.2.18. LanguageSetter.....	41
3.3.2.19. PieceDisplay.....	41
3.3.2.20. PlayerSetter.....	41
3.3.2.21. SizeSetter.....	42
3.3.2.22. TileConverter.....	42
3.4. Paquet d'utilitats.....	42
3.4.1. Direction.....	42
3.4.2. IRand.....	42

3.4.3. Pair.....	42
3.4.4. Rand.....	42
3.4.5. Vector2.....	42
4. Descripció dels tests.....	43
4.1. TestDrawActionMaker.....	43
4.1.1. successSwap.....	43
4.1.2. piecesToSwapNull.....	43
4.1.3. notEnoughBagPieces.....	43
4.1.4. multipleBagPieces.....	43
4.2. TestPlaceActionMaker.....	44
4.2.1. placePieceSendsUpdateViewCallback.....	44
4.2.2. placePieceThrowsExceptionIfWordDoesNotExist.....	44
4.2.3. placePieceThrowsExceptionIfMovementIsOutsideOfBoard.....	44
4.2.4. placePieceThrowsExceptionIfContainsCombinedWordThatDoesNotExist.....	45
4.2.5. placePiecesSubtractsPiecesFromPlayer.....	45
4.2.6. placedPiecesInHandGetReplacedByBagPieces.....	45
4.2.7. placedPiecesGetPlacedUpdatesBoard.....	46
4.2.8. continuousPlacesCorrectlyPlacedOnBoard.....	46
4.2.9. placePieceThrowsExceptionIfNotConnectedToOtherWord.....	46
4.2.10. placeWordWithBlankPiece.....	47
4.2.11. placeFirstOutsideOfCenter.....	47
4.3. TestSkipActionMaker.....	47
4.3.1. skipRunTriggersNextTurn.....	47
4.4. TestAI.....	48
4.4.1. englishAIEmptyBoard.....	48
4.4.2. catalanAIEmptyBoardNY.....	48
4.4.3. catalanAIEmptyBoardLGeminada.....	48
4.4.4. catalanAIEmptyBoardNYEndOfWord.....	49
4.4.5. catalanAIEmptyBoardLGeminadaEndOfWord.....	49
4.4.6. spanishAIEmptyBoardRR.....	49
4.4.7. spanishAIEmptyBoardRREndOfWord.....	50
4.4.8. spanishAIEmptyBoardLL.....	50
4.4.9. spanishAIEmptyBoardLLEndOfWord.....	50
4.4.10. spanishAIEmptyBoardCH.....	51
4.4.11. spanishAIEmptyBoardCHEndOfWord.....	51
4.4.12. englishAIHorizontal.....	51
4.4.13. catalanAIHorizontalNYRETRUNSMAXMOVEPOINTS.....	52
4.4.14. catalanAIHorizontalLGeminada.....	52
4.4.15. spanishAIHorizontalRR.....	52
4.4.16. spanishAIHorizontalLL.....	53
4.4.17. spanishAIHorizontalCH.....	53
4.4.18. englishAIHorizontalwithBlank.....	53
4.4.19. catalanAIEmptyBoardNYwithBlank.....	54
4.4.20. spanishAIEmptyBoardRRwithBlank.....	54

4.4.21. englishAlVertical.....	54
4.4.22. englishAladjacent.....	55
4.4.23. dontRepeatWords.....	55
4.5. TestAnchors.....	55
4.5.1. initialAnchor.....	55
4.5.2. updateAnchorHor.....	55
4.5.3. updateAnchorVer.....	55
4.5.4. rotateAnchors.....	56
4.6. TestCrossChecks.....	56
4.6.1. createCrossChecks.....	56
4.6.2. updateCrossChecks.....	56
4.6.3. rotateCrossChecks.....	56
4.7. TestBoard.....	57
4.7.1. standardBoardHasSize15.....	57
4.7.2. superBoardHasSize21.....	57
4.7.3. jrBoardHasSize11.....	57
4.7.4. boardIsEmptyWhenNoPiecesPlaced.....	57
4.7.5. boardIsNotEmptyWhenPiecesPlaced.....	57
4.7.6. placePieceAddsItToBoard.....	57
4.7.7. placePiecePlacesCorrectPiece.....	57
4.7.8. readEmptyCellFromBoardReturnsNull.....	58
4.7.9. centerTileIsNotCorrectWhenNotCenter.....	58
4.7.10. centerTileIsCorrectInStandardBoard.....	58
4.7.11. centerTileIsCorrectInSuperBoard.....	59
4.7.12. centerTileIsCorrectInJuniorBoard.....	59
4.7.13. emptyCellReturnsTrueWhenCellIsEmpty.....	59
4.7.14. emptyCellReturnsFalseWhenCellIsNotEmpty.....	59
4.7.15. isPremiumTileReturnsTrueWhenCellIsPremium.....	59
4.7.16. isPremiumTileReturnsFalseWhenCellIsNotPremium.....	59
4.7.17. rotateBoardRotatesPiece.....	59
4.7.18. rotateBoardRotatesWord.....	60
4.8. TestPointCalculator.....	60
4.8.1. calculateStandardPointsWithNoPremium.....	60
4.8.2. calculatePointsWhenPlacingSinglePiece.....	60
4.8.3. calculatePointsWhenPlacingSinglePiece AlreadyOnBoard.....	60
4.8.4. calculateStandardPointsWithPremiumLetter.....	60
4.8.5. calculateStandardPointsWithPremiumWord.....	61
4.8.6. calculateBingoBonus.....	61
4.8.7. calculatePointsForTwoWordsSharingOnePiece.....	61
4.8.8. calculatePointsForTwoWordsSharing OnePieceNotPlaced.....	61
4.8.9. calculatePointsForThreeWordsWith DifferentMultipliers.....	61
4.8.10. calculatePointsForThreeWordsWith DifferentMultipliersNotPlaced.....	62
4.8.11. calculatePointsForThreeWordsWith Gaps.....	62
4.8.12. calculatePointsForThreeWordsWithGapsNotPlaced.....	62

4.8.13. calculatePointsForThreeWords.....	62
4.8.14. calculatePointsForThreeWordsNotPlaced.....	63
4.9. TestPresentPiecesWordCompleter.....	63
4.9.1. checkWordCompleterReturnsExtendedWord.....	63
4.9.2. checkWordCompleterReturnsCrossedWord.....	63
4.9.3. checkWordCompleterReturnsTwoCompletedWordsWhenPlacingOnePiece... 63	
4.9.4. checkWordCompleterReturnsTwoCompletedWordsWhenCompletingMorePieces... 64	
4.9.5. checkWordCompleterReturnsAllAdjacentPiecesOfTwoParallelWords.....	64
4.10. TestWordGetter.....	64
4.10.1. getVerticalWord.....	64
4.10.2. getVerticalWordWithTwoWordsInline.....	64
4.10.3. getHorizontalWord.....	65
4.10.4. getHorizontalWordWithTwoWordsInline.....	65
4.10.5. getHorizontalWordThatWasAlreadyPlaced.....	65
4.10.6. getHorizontalWordWithPreviousWordHavingSameCharacter.....	65
4.11. TestWordPlacer.....	66
4.11.1. placeWordVerticalPlacesWord.....	66
4.11.2. placeWordHorizontalPlacesWord.....	66
4.11.3. noPlaceActionDoesntUpdateBoard.....	66
4.11.4. placeActionUpdatesBoard.....	66
4.12. TestGamesPlayed.....	67
4.12.1. testRunReturnsCorrectGamesPlayedSorted.....	67
4.13. TestGamesWon.....	67
4.13.1. testRunReturnsCorrectGamesWonSorted.....	67
4.14. TestMaxScore.....	67
4.14.1. testRunReturnsCorrectMaxScoreSorted.....	67
4.15. TestTotalScore.....	68
4.15.1. testRunReturnsCorrectTotalScoreSorted.....	68
4.16. TestWinRate.....	68
4.16.1. testRunReturnsCorrectWinRatesSorted.....	68
4.17. TestDrawParser.....	68
4.17.1. testParseSingleLetter.....	68
4.17.2. testParseTwoLetters.....	68
4.17.3. testParseSpecialDigraph.....	69
4.17.4. testParseBlankTile.....	69
4.18. TestMovementBoundsChecker.....	69
4.18.1. testExactJuniorHorizontalBounds.....	69
4.18.2. testExactJuniorVerticalBounds.....	69
4.18.3. testMoreThanJuniorHorizontalBounds.....	69
4.18.4. testMoreThanJuniorVerticalBounds.....	70
4.18.5. testExactStandardHorizontalBounds.....	70
4.18.6. testExactStandardVerticalBounds.....	70
4.18.7. testMoreThanStandardHorizontalBounds.....	70

4.18.8. testMoreThanStandardVerticalBounds.....	71
4.18.9. testExactSuperHorizontalBounds.....	71
4.18.10. testExactSuperVerticalBounds.....	71
4.18.11. testMoreThanSuperHorizontalBounds.....	71
4.18.12. testMoreThanSuperVerticalBounds.....	72
4.19. TestMovementCleaner.....	72
4.19.1. test1PieceCleanerVertical.....	72
4.19.2. test2PiecesCleanerVertical.....	72
4.19.3. test3PiecesCleanerVertical.....	72
4.19.4. testAllPiecesCleanerVertical.....	73
4.19.5. test1PiecesCleanerHorizontal.....	73
4.19.6. test2PiecesCleanerHorizontal.....	73
4.19.7. test3PiecesCleanerHorizontal.....	73
4.19.8. testAllPiecesCleanerHorizontal.....	74
4.20. TestMovementParser.....	74
4.20.1. testMakeMoveReturnsCorrectWord.....	74
4.20.2. testMakeMoveReturnsCorrectPosition.....	74
4.20.3. testMakeMoveReturnsCorrectPositionInverted.....	74
4.20.4. testMakeMoveReturnsCorrectDirectionHorizontal.....	74
4.20.5. testMakeMoveReturnsCorrectDirectionVertical.....	74
4.21. TestPlayer.....	75
4.21.1. addScoreUpdatesPlayerScore.....	75
4.21.2. playerScoreEqualsZero.....	75
4.21.3. scoreAddsProperly.....	75
4.22. TestBag.....	75
4.22.1. bagDefaultsToEmpty.....	75
4.22.2. bagAddsOnePiece.....	75
4.22.3. bagGetsOnePiece.....	75
4.22.4. bagThrowsErrorIfTryGettingWhenEmpty.....	76
4.22.5. bagThrowsErrorIfTryAddingNullPiece.....	76
4.23. TestBagFiller.....	76
4.23.1. bagIsFilled.....	76
4.24. TestDAWG.....	76
4.24.1. createDAWGproperly.....	76
4.24.2. addWord.....	76
4.24.3. addMultipleWords.....	77
4.25. TestHandFiller.....	77
4.25.1. tryFillWithEmptyBagThrowsException.....	77
4.25.2. tryFillWithNotEnoughPiecesInBag ThrowsException.....	77
4.25.3. fillHandFillsPlayerCorrectly.....	77
4.25.4. fillHandFillsAllPlayersCorrectly.....	78
4.26. TestPiece.....	78
4.26.1. pieceCreator.....	78
4.26.2. pieceComparator.....	78

4.26.3. pieceBlank.....	78
4.26.4. setLetterBlankPiece.....	78
4.27. TestPieceDrawer.....	79
4.27.1. swap1piece.....	79
4.27.2. swap2pieces.....	79
4.27.3. throwExceptionIfNotEnoughPiecesInBag.....	79
4.28. TestPieceGenerator.....	79
4.28.1. whenPassingDPieceStringGenerateDPiece.....	79
4.28.2. whenPassingBlankPieceStringGenerateBlankPiece.....	80
4.28.3. whenPassingCPieceStringGenerateCPiece.....	80
4.28.4. whenPassingDPieceStringGenerateDValue.....	80
4.28.5. whenPassingEPieceStringGenerateEValue.....	80
4.28.6. whenPassingDPieceStringGenerateDCount.....	80
4.28.7. whenPassingEPieceStringGenerateECount.....	81
4.28.8. parseMultiplePieces.....	81
4.29. TestPiecesConverter.....	81
4.29.1. getLPieceFromLCharacter.....	81
4.29.2. getRPieceFromRCharacter.....	81
4.29.3. getPiecesFromWord.....	81
4.29.4. getLgeminadaPiecesFromLgeminadaCharacterCAT.....	82
4.29.5. getNYPiecesFromNYCharacterCAT.....	82
4.29.6. getRRPiecesFromRRCharacterCAST.....	82
4.29.7. getLLPiecesFromLLCharacterCAST.....	82
4.29.8. getCHPiecesFromCHCharacterCAST.....	82
4.29.9. getPieceAsBlankPieceWhenCharacterIsLowercase.....	83
4.29.10. getLgeminadaBlankPieceFromLgeminadaCharacterCAT.....	83
4.29.11. getNYBlankPieceFromNYCharacterCAT.....	83
4.29.12. getRRBlankPieceFromRRCharacterCAST.....	83
4.29.13. getLLBlankPieceFromLLCharacterCAST.....	83
4.29.14. getCHBlankPieceFromCHCharacterCAST.....	84
4.29.15. getPieceWithDictionaryConvertsItWithProperScore.....	84
4.29.16. getSpecialCatalanPieceWithDictionaryConvertsItWithProperScore.....	84
4.29.17. getSpecialSpanishPieceWithDictionaryConvertsItWithProperScore.....	84
4.29.18. getSpecialSpanishBlankPieceWithDictionaryConvertsItWithNoScore.....	85
4.29.19. getSpecialCatalanBlankPieceWithDictionaryConvertsItWithNoScore.....	85
4.30. TestPiecesConverterFactory.....	85
4.30.1. generatePiecesConverterWithEnglishDictionary.....	85
4.30.2. generatePiecesConverterWithCatalanDictionary.....	85
4.30.3. generatePiecesConverterWithSpanishDictionary.....	86
4.30.4. newPiecesConverterParsingWithCorrectLanguageScore.....	86
4.30.5. newPiecesConverterParsingWithMultipleScores.....	86
4.31. TestPiecesInHandGetter.....	86
4.31.1. get1Piece.....	86
4.31.2. get1BlankPiece.....	87

4.31.3. get1BlankPieceThrowsExceptionIfPlayerDoesNotHavePiece.....	87
4.31.4. rollbackWhenOnePieceNotFound.....	87
4.31.5. get1PieceRemovesOnePieceFromBag.....	87
4.31.6. get2Pieces.....	88
4.32. TestPiecesInHandVerifier.....	88
4.32.1. getPieceABInHandAB.....	88
4.32.2. getPieceABInHandABCD.....	88
4.32.3. getPieceNYInHandABNY.....	88
4.32.4. getPieceCHInHandABCH.....	89
4.32.5. getPieceLgeminadaInHandABLgeminada.....	89
4.32.6. getPieceRRInHandABRR.....	89
4.32.7. getPieceNYCHRRInHandABNYCHRR.....	89
4.33. TestSceneObject.....	90
4.33.1. createSceneObjectCallsConstructor.....	90
4.33.2. createSceneObjectIsEnabledByDefault.....	90
4.33.3. disableObjectDisablesIt.....	90
4.33.4. sceneProcessCallsObjectProcessMethod.....	90
4.33.5. sceneProcessDoesNotCallObjectProcessMethodWhenDisabled.....	91
4.33.6. destroyObjectCallsDetachMethod.....	91
4.34. TestEndgame.....	91
4.34.1. TwoPlayersEndgameDueToSkipLimitReached.....	91
4.34.2. TwoPlayersNoEndgameDueToSkipLimitReached.....	91
4.34.3. TwoPlayersEndgameDueToSomeoneWithEmptyHand.....	92
4.34.4. ThreePlayersEndgameDueToSkipLimitReached.....	92
4.34.5. ThreePlayersNoEndgameDueToSkipLimitReached.....	92
4.34.6. ThreePlayersEndgameDueToSomeoneWithEmptyHand.....	92
4.34.7. FourPlayersEndgameDueToSkipLimitReached.....	93
4.34.8. FourPlayersNoEndgameDueToSkipLimitReached.....	93
4.34.9. FourPlayersEndgameDueToSomeoneWithEmptyHand.....	93
5. Algorismes utilitzats.....	94
5.1. DAWG.....	94
5.2. Scrabble.....	95
6. Estructures de dades.....	97
6.1. Gestió d'escenes: SceneManager, Scene i SceneObject.....	97
6.1.1. SceneManager.....	97
6.1.2. Scene.....	97
6.1.3. SceneObject.....	97
6.2. Lectura de diccionaris i fitxes: DictionaryReader i PiecesReader.....	99
6.2.1. LocaleReader.....	99
6.2.2. PiecesReader.....	99
6.2.3. Pieces.....	99
6.3. Emmagatzematge i gestió de puntuacions: Score, Leaderboard i controladors.....	100
6.3.1. Score.....	100
6.3.2. Leaderboard.....	100

6.3.3. Controladors.....	100
6.4. Emmagatzematge dels moviments.....	101
7. Relació de classes.....	102
7.1. Capa de dades.....	102
7.1.1. Mòdul board.....	102
7.1.2. Mòdul crosschecks.....	102
7.1.3. Mòdul DAWG.....	102
7.1.4. Mòdul exceptions.....	102
7.1.5. Mòdul leaderboard.....	102
7.1.6. Mòdul pieces.....	102
7.1.7. Mòdul properties.....	102
7.2. Capa de domini.....	103
7.2.1. Mòdul actionmaker.....	103
7.2.2. Mòdul AI.....	103
7.2.3. Mòdul board.....	103
7.2.4. Mòdul DAWG.....	103
7.2.5. Mòdul exceptions.....	103
7.2.6. Mòdul game.....	104
7.2.7. Mòdul leaderboard.....	104
7.2.8. Mòdul movement.....	104
7.2.9. Mòdul pieces.....	104
7.2.10. Mòdul turns.....	104
7.3. Capa de presentació.....	105
7.3.1. Mòdul localization.....	105
7.3.2. Mòdul scenes.....	105
7.3.2.1. Mòdul excepcions.....	105
7.3.3. Mòdul terminal.....	105
7.3.3.1. Mòdul actionmaker.....	105
7.3.3.2. Mòdul movements.....	105
7.3.3.3. Mòdul players.....	105
7.3.3.4. Mòdul scenes.....	105
7.3.3.5. Mòdul utils.....	105
7.4. Testing.....	106
7.4.1. Mòdul actionmaker.....	106
7.4.2. Mòdul AI.....	106
7.4.3. Mòdul board.....	106
7.4.4. Mòdul leaderboard.....	106
7.4.5. Mòdul movement.....	106
7.4.6. Mòdul pieces.....	106
7.4.7. Mòdul scenes.....	107
7.4.8. Mòdul stubs.....	107
7.4.9. Mòdul turns.....	107
7.5. Paquet d'utilitats.....	107

1. Decisions preses

1.1. Capa de dades

En aquesta primera entrega, la **capa de dades** s'ha dissenyat amb l'objectiu de contenir **únicament dades abstractes** i una **lògica mínima**, centrada en la representació de l'estat del joc i les seves estructures fonamentals (com el tauler, les fitxes, els jugadors, etc.).

Aquesta elecció s'ha fet de manera conscient per **separar la lògica de negoci de qualsevol dependència tecnològica**, com ara la persistència o la interfície d'usuari. D'aquesta manera, la capa de dades es manté **lleugera, testejaable i independent**, facilitant l'evolució del projecte a llarg termini.

L'objectiu és que, de cara a la **tercera entrega**, es pugui integrar la **persistència de dades** (ja sigui mitjançant arxius, bases de dades o altres sistemes) **sense modificar la lògica central de la capa de dades**. Això s'assolirà mitjançant la incorporació de capes o components addicionals responsables exclusivament de la persistència, els quals interactuaran amb la capa de dades a través de mecanismes ben definits.

1.2. Capa de domini

La **capa de domini** del projecte ha estat dissenyada seguint els principis de l'arquitectura **MVP (Model-View-Presenter)**¹, actuant com a pont entre la capa de presentació i la capa de dades. En aquesta capa, es gestiona tota la **lògica de coordinació i control**, separant completament la interfície d'usuari de la lògica de negoci.

A nivell estructural:

- El **domini es comunica directament amb la capa de dades**, cridant els mètodes necessaris per modificar i consultar l'estat del joc.
- La comunicació amb la vista es fa mitjançant **inversió de dependències**, és a dir, la vista implementa una interfície definida pel **domini**, de manera que aquest **no depèn de cap implementació concreta** de la vista. Això permet canviar la vista (per exemple, passar d'una versió textual a una amb Swing) **sense afectar la lògica del domini**.

A més, s'ha seguit el principi de **responsabilitat única**, de manera que **cada classe dins de la capa de domini representa una acció concreta** dins del programa (com ara jugar una fitxa, passar torn, validar una paraula, etc.). Aquesta organització millora la **llegibilitat, mantenibilitat i testabilitat** del codi, ja que cada classe té una funció clara i acotada dins del flux del joc.

¹ Fowler, M. (2006, 18 juliol). GUI Architectures. [martinfowler.com. https://martinfowler.com/eaaDev/uiArchs.html#Model-view-presentermvp](https://martinfowler.com/eaaDev/uiArchs.html#Model-view-presentermvp)

Aquesta estructura modular també facilita l'evolució del sistema, que permet afegir noves funcionalitats o modificar les existents sense introduir complexitat excessiva ni crear dependències innecessàries entre components.

1.3. Capa de presentació

Primer de tot, entenem com a vista **tot allò relacionat amb la interacció amb l'usuari i/o el sistema**, ja sigui a través d'una interfície gràfica, la lectura i escriptura per terminal o fins i tot la lectura i escriptura d'arxius del sistema. Aquesta definició amplia el concepte tradicional de vista i ens permet aplicar els mateixos principis arquitectònics a qualsevol forma de presentació o entrada d'informació per part de l'usuari.

És important tenir present que l'objectiu principal d'aquesta primera entrega és disposar d'una capa de domini funcional, no pas d'una vista definitiva. Per això, la vista temporal que s'ha implementat no té la funció de validar el correcte funcionament global del programa, sinó que serveix com a eina per visualitzar i assajar la comunicació entre capes, especialment entre la *capa presentació* i el *domini*. La verificació del comportament correcte del sistema es fa mitjançant proves unitàries, que garanteixen que la lògica del domini es comporta tal com s'espera, de manera independent de la presentació.

Aquesta vista temporal no utilitza encara la llibreria Swing, però ja incorpora una interfície de comunicació amb la capa de domini. La implementació segueix els principis de l'arquitectura **MVP**, en particular, es basa en el **patró vista passiva**², assegurant que la vista no conté lògica de negoci i que tota la interacció amb la *capa de dades* es gestiona a través de la *capa de domini*. Aquesta aproximació permet mantenir una arquitectura modular, escalable i fàcilment testable.

La interfície definida entre la *capa presentació* i la *capa domini* simula el comportament que oferirà la futura vista gràfica basada en Swing. Això permet desenvolupar i provar el sistema de forma iterativa, avançant en les capes de Domini i Dades sense quedar condicionats per la part gràfica.

Quan arribi el moment d'introduir la implementació gràfica amb Swing (prevista per a la tercera entrega), només caldrà substituir la vista temporal per una nova implementació que reutilitzi la mateixa interfície. Això minimitzarà l'impacte en el codi existent i evitarà haver de modificar la lògica de les capes inferiors, en línia amb el **principi de separació de responsabilitats** propi del patró MVP i el patró vista passiva.

A més, aquesta estratègia facilita les proves unitàries, ja que les interfícies definides per a la comunicació amb la vista temporal poden actuar com a mock o stub durant el desenvolupament inicial, cosa que permet validar de forma aïllada el comportament del sistema.

² Fowler, M. (2006, 18 juliol). Passive View. [martinfowler.com. https://martinfowler.com/eaDev/PassiveScreen.html](https://martinfowler.com/eaDev/PassiveScreen.html)

1.4. Gestió de dependències

Per a la gestió de les dependències entre components, s'ha optat per utilitzar el principi **d'injecció de dependències**³ com a mètode principal. Aquest enfocament consisteix a **proporcionar explícitament les dependències** que necessita cada classe en el moment de la seva creació, en lloc de crear-les internament.

El joc s'ha estructurat en diferents **escenes**, cadascuna de les quals representa un estat o fase concreta del joc (per exemple, escena de menú, escena de partida, etc.). Aquestes escenes no només encapsulen la lògica i vista associada a un moment concret del joc, sinó que també actuen com a **punt central de resolució de dependències**.

Tot i que no s'utilitza cap framework d'injecció automàtica, la **injecció es fa de manera manual** durant la inicialització de cada escena, seguint l'esquema següent:

- La **capa de domini** rep com a dependències instàncies de la **capa de dades** i/o interfícies de la **capa de presentació**.
- La **capa de presentació** rep instàncies de la **capa de domini** com a referència per delegar-hi les accions de l'usuari.
- Qualsevol component que necessiti altres serveis o col·laboradors els rep per constructor.

Aquesta estratègia presenta diversos avantatges:

- **Centralitza la configuració de cada escena**, facilitant-ne el control i la modificació.
- Permet mantenir les dependències **desacoblades i explícites**, millorant la llegibilitat i la mantenibilitat del codi.
- **Facilita el testing**, ja que el fet d'injectar dependències manualment permet substituir components reals per simulacions durant les proves.
- Respecta el principi de **responsabilitat única**, ja que cada escena és responsable de construir i enllaçar els seus propis components.
- **Respecta el principi d'inversió de dependències**, mantenint les dependències apuntant cap a abstraccions i no cap a implementacions concretes.

Aquest patró proporciona una base sòlida i flexible per escalar el projecte en entregues futures, per exemple, afegint nous tipus d'escenes o integrant serveis de persistència sense trencar la lògica actual.

³ Fowler, M. (2004, 23 gener). Inversion of Control Containers and the Dependency Injection pattern. martinofowler.com. <https://martinfowler.com/articles/injection.html>

1.5. Testing

Element ^	Class, %	Method, %	Line, %	Branch, %
edu.upc.prop.scrabble	94% (66/70)	94% (289/306)	95% (1333/1399)	77% (619/802)
data	86% (20/23)	89% (106/118)	93% (426/454)	76% (75/98)
board	83% (5/6)	89% (33/37)	97% (278/285)	95% (23/24)
crosschecks	100% (4/4)	100% (20/20)	90% (40/44)	72% (16/22)
dawg	100% (2/2)	77% (14/18)	80% (33/41)	37% (6/16)
exceptions	100% (2/2)	100% (2/2)	100% (2/2)	100% (0/0)
leaderboard	100% (2/2)	100% (4/4)	100% (4/4)	100% (0/0)
pieces	100% (2/2)	100% (13/13)	96% (28/29)	77% (14/18)
properties	0% (0/2)	0% (0/3)	0% (0/5)	100% (0/0)
anchors	100% (1/1)	100% (7/7)	100% (10/10)	100% (2/2)
movement	100% (1/1)	100% (5/5)	100% (5/5)	100% (0/0)
player	100% (1/1)	88% (8/9)	89% (26/29)	87% (14/16)
domain	100% (43/43)	100% (162/162)	96% (853/880)	77% (528/684)
actionmaker	100% (3/3)	100% (13/13)	100% (58/58)	100% (22/22)
ai	100% (4/4)	100% (27/27)	95% (210/221)	66% (185/280)
board	100% (5/5)	100% (31/31)	98% (156/159)	94% (79/84)
crosschecks	100% (1/1)	100% (10/10)	100% (62/62)	87% (42/48)
dawg	100% (2/2)	100% (11/11)	94% (49/52)	84% (22/26)
exceptions	100% (5/5)	100% (5/5)	100% (5/5)	100% (0/0)
game	100% (1/1)	100% (2/2)	100% (13/13)	90% (9/10)
leaderboard	100% (7/7)	100% (12/12)	100% (45/45)	100% (30/30)
movement	100% (2/2)	100% (8/8)	94% (50/53)	70% (21/30)
pieces	100% (9/9)	100% (35/35)	95% (156/163)	82% (79/96)
turns	100% (3/3)	100% (6/6)	100% (27/27)	100% (10/10)
AnchorUpdater	100% (1/1)	100% (2/2)	100% (22/22)	60% (29/48)
presenter.scenes	75% (3/4)	80% (21/26)	83% (54/65)	80% (16/20)
exceptions	0% (0/1)	0% (0/1)	0% (0/1)	100% (0/0)
Scene	100% (1/1)	100% (5/5)	83% (20/24)	80% (8/10)
SceneManager	100% (1/1)	80% (8/10)	85% (24/28)	80% (8/10)
SceneObject	100% (1/1)	80% (8/10)	83% (10/12)	100% (0/0)

Figura 1: Cobertura dels tests unitaris

En aquesta primera entrega del projecte, centrada exclusivament en la implementació del domini, hem optat per desenvolupar usant únicament tests unitaris, sense incloure jocs de prova end-to-end per terminal. Aquesta decisió s'ha pres principalment per dues raons:

1. **Focalització en la lògica de domini:** L'objectiu principal d'aquesta fase era garantir la correcció i robustesa de la lògica de negoci. Els tests unitaris ens han permès verificar de forma exhaustiva i precisa el comportament de les funcions i classes del domini de manera aïllada, amb una cobertura molt alta⁴. Això ens ha proporcionat confiança en la qualitat del codi desenvolupat.
2. **Limitada rellevància dels jocs de prova en aquest punt del projecte:** Atès que en aquesta etapa encara no s'ha desenvolupat la part d'interacció amb l'usuari, els tests end-to-end mitjançant executables o jocs de prova no haurien aportat un valor afegit significatiu. En aquesta fase, aquests tests haurien estat artificials o massa limitats per reflectir casos d'ús reals.

Tenint en compte això, considerem que hem aplicat els recursos de forma eficient centrant-nos en els tests unitaris, i ens reservem la integració de proves end-to-end per etapes posteriors del projecte, quan realment es pugui validar el sistema de manera global.

⁴ Veure figura 1 del document

2. Casos d'ús

2.1. Diagrama de casos d'ús

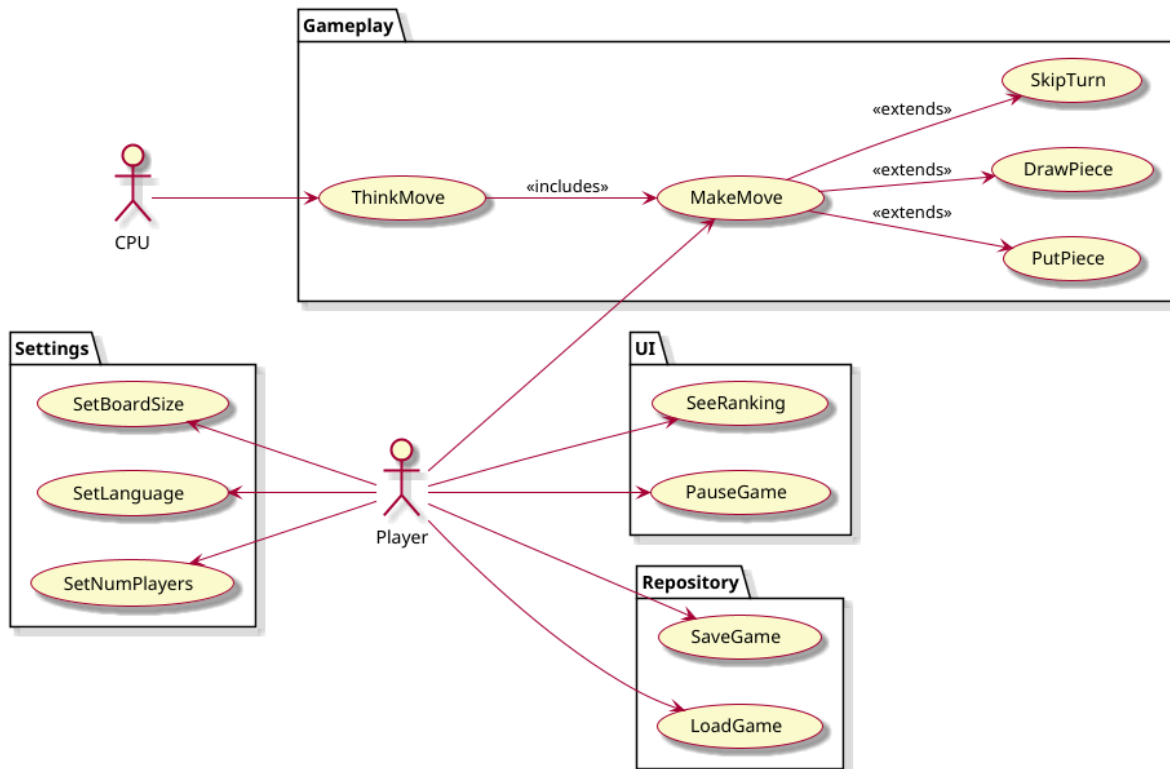


Figura 2: Imatge del diagrama de casos d'ús

2.2. Actors

2.2.1. Player

Un jugador humà que interactua amb el videojoc.

2.2.2. CPU

Un oponent controlat per l'ordinador que competeix contra el jugador humà.

2.3. Descripció individual dels casos d'ús

2.3.1. SeeRanking

El jugador pot consultar el rànquing global del joc i aplicar diversos filtres per personalitzar la visualització dels resultats. Els filtres disponibles inclouen:

- Nombre total de partides jugades
- Nombre de partides guanyades
- Màxima puntuació obtinguda
- Puntuació acumulada al llarg de totes les partides
- Ratxa de partides guanyades

2.3.2. PauseGame

El jugador pot interrompre la partida en qualsevol moment i accedir al menú de pausa. Aquest menú ofereix diferents opcions per reprendre el joc, guardar-lo, o sortir a la pantalla principal.

2.3.3. SetBoardSize

El jugador pot triar la mida del tauler per a la partida següent. Les mides disponibles són:

- **Junior (11x11):** Tauler petit, ideal per a principiants o partides ràpides.
- **Standard (15x15):** Tauler estàndard, el més comú per a partides tradicionals.
- **Super (21x21):** Tauler gran, per a partides més llargues i complexes.

2.3.4. SetLanguage

El jugador pot seleccionar l'idioma del joc, fet que influirà en el diccionari i les fitxes utilitzades en la partida. Els idiomes disponibles són:

- Català
- Castellà
- Anglès

2.3.5. SetNumPlayers

El jugador pot escollir el nombre de participants per a la pròxima partida, amb opcions per jugar entre 2 i 4 jugadors. També pot decidir si els jugadors seran controlats per l'ordinador, assignant un nom a cadascun d'ells.

2.3.6. ThinkMove

La CPU analitza la situació actual al tauler i pensa quin és el millor moviment a realitzar, tenint en compte les fitxes disponibles i les condicions del joc.

2.3.7. MakeMove

Tant el jugador com la CPU poden realitzar un moviment durant el seu torn. Els tipus de moviment són:

- Col·locar una fitxa (PutPiece)
- Robar una fitxa (DrawPiece)
- Passar el torn (SkipTurn)

2.3.8. PutPiece

El jugador o la CPU poden col·locar una fitxa al tauler. Aquesta acció atorga els punts corresponents segons les regles del joc. Si el jugador ha col·locat fitxes, roba tantes fitxes com les que ha jugat, sempre que n'hi hagi prou al sac de fitxes. El torn es finalitza automàticament després de fer aquest moviment.

2.3.9. DrawPiece

El jugador o la CPU poden intercanviar qualsevol nombre de fitxes per noves fitxes del sac, sempre que hi hagi prou fitxes disponibles. El torn s'acaba automàticament després d'aquesta acció.

2.3.10. SkipTurn

El jugador o la CPU poden optar per no fer cap moviment durant el seu torn. Això inclou no col·locar fitxes ni robar-ne del sac. El torn s'acaba immediatament després de saltar-lo.

2.3.11. SaveGame

El jugador pot guardar el progrés de la partida en qualsevol moment. Això crea un arxiu de guardat amb tota la informació necessària per reprendre el joc més endavant, mantenint l'estat de la partida tal com estava.

2.3.12. LoadGame

El jugador pot carregar una partida prèviament guardada, reiniciant la partida amb el mateix estat que tenia en el moment del guardat, incloent-hi la posició de les fitxes, les puntuacions i les opcions de joc.

3. Classes

A causa de l'aplicació del **Principi de Responsabilitat Única**, hem dissenyat el sistema amb múltiples classes, cadascuna encarregada d'una única responsabilitat. Aquesta decisió facilita el manteniment, la reutilització i la prova unitària del codi, ja que cada classe té un propòsit ben definit.

Com a conseqüència, el **diagrama de classes complet** pot resultar visualment complex i dens, dificultant la seva comprensió en una sola imatge. Per aquest motiu, i amb l'objectiu de millorar-ne la llegibilitat i claredat, hem optat per **dividir el diagrama en diferents seccions** corresponents a les capes i components del sistema:

- **Capa de dades**
- **Capa de domini**
- **Capa de presentació**

Aquesta divisió ens permet presentar les relacions i responsabilitats de les classes dins del seu context específic, ajudant a entendre millor l'arquitectura modular del projecte.

Tot i això, per tal d'oferir una visió global del sistema, també s'adjunta una **imatge amb el diagrama de classes complet** en un arxiu a part, que mostra totes les dependències i interaccions entre els components.

3.1. Capa de dades

3.1.1. Diagrama

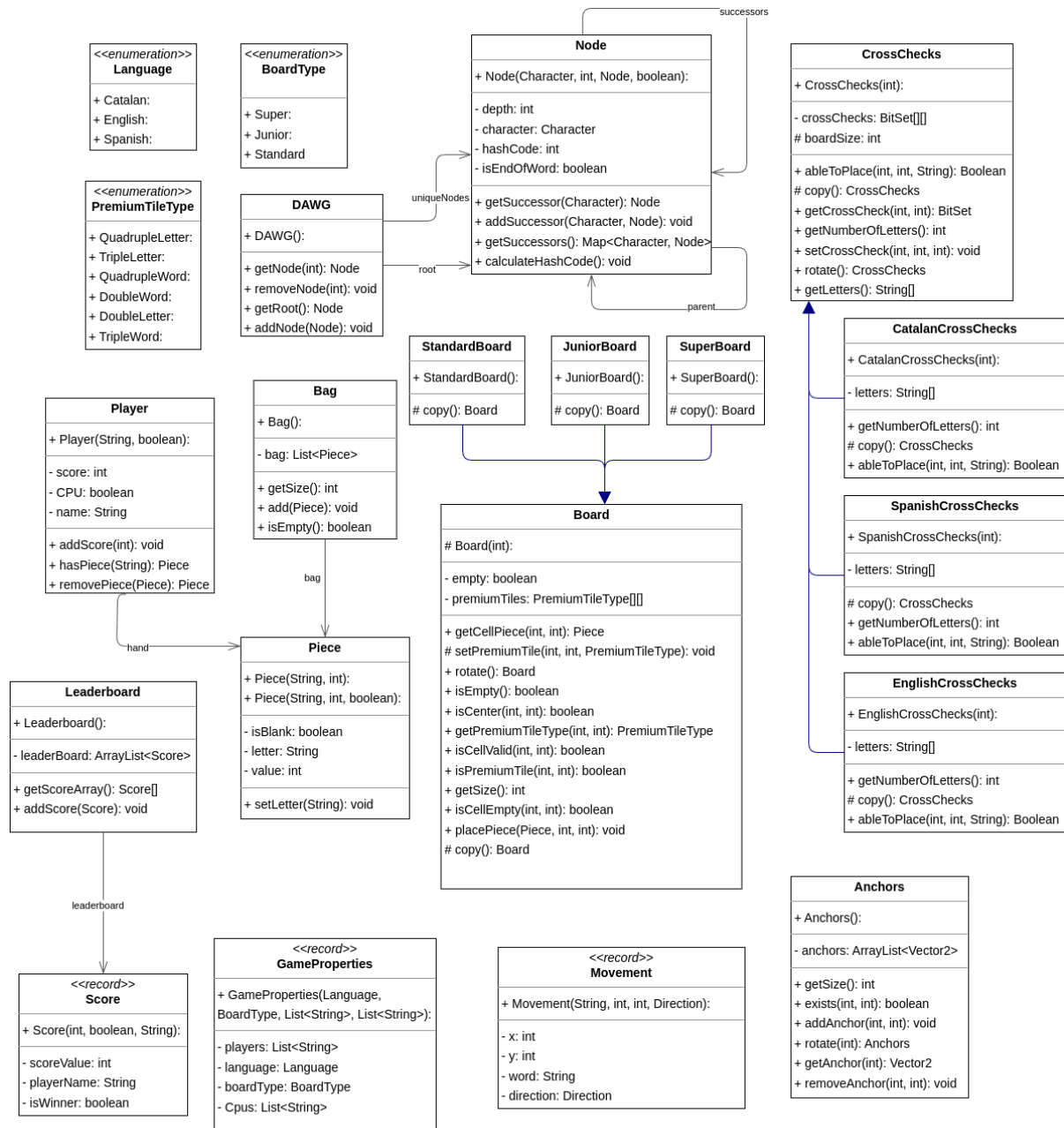


Figura 3: Imatge del diagrama de la capa de dades

3.1.2. Descripció

3.1.2.1. Board

Representa el tauler del joc Scrabble. Manté les fitxes de lletres col·locades i la informació sobre les caselles de puntuació especial (premium). Aquesta classe proporciona funcionalitats bàsiques com col·locar fitxes, comprovar l'estat de les cel·les i girar el tauler.

Funcionalitats principals:

- **Col·locar fitxes:** Permet col·locar una fitxa de lletra en una posició especificada del tauler.
- **Comprovar cel·les:** Verifica si una cel·la està buida o ocupa el centre del tauler, que ha de ser cobert pel primer mot jugat.
- **Puntuació especial:** Identifica i gestiona les caselles de puntuació premium (doble/triple de paraula o lletra).
- **Girar el tauler:** Retorna una còpia del tauler girat 90 graus en el sentit de les agulles del rellotge.
- **Comprovar validesa:** Verifica si les coordenades d'una cel·la són vàlides dins del tauler.

Aquesta classe és abstracta, la qual cosa vol dir que ha de ser estesa per altres classes que proporcionin la seva implementació concreta.

3.1.2.2. BoardType

Defineix els tipus disponibles de tauler per al joc Scrabble. Cada tipus pot tenir una mida, disposició o conjunt de regles diferents:

- **Junior:** Versió simplificada de mida 11x11, dissenyada per a jugadors més joves.
- **Standard:** Tauler oficial de mida 15x15, utilitzat en el Scrabble clàssic.
- **Super:** Variant ampliada de mida 21x21.

3.1.2.3. JuniorBoard

Representa un tauler de Scrabble configurat per a la variant Junior, amb una disposició 11x11 i posicionament específic de les caselles de puntuació especial (premium).

Funcionalitats principals:

- **Tauler 11x11:** Inicia un tauler de mida 11x11 dissenyat per a jugadors més joves.
- **Caselles premium:** Configura les caselles de puntuació especial, incloent-hi les de **doble paraula**, **triple paraula**, **doble lletra** i **triple lletra** en posicions predefinides.
- **Còpia:** Implementa la creació d'una còpia del tauler actual per permetre la duplicació de l'estat del joc.

Aquesta classe estén la classe **Board** i personalitza les caselles de puntuació premium segons les regles de la variant Junior.

3.1.2.4. PremiumTileType

Representa els tipus de caselles premium en un tauler de Scrabble. Aquestes caselles multipliquen el valor de la lletra col·locada o de tota la paraula.

Tipus de caselles premium:

- **QuadrupleWord**: Multiplica la puntuació de la paraula per 4.
- **TripleWord**: Multiplica la puntuació de la paraula per 3.
- **DoubleWord**: Multiplica la puntuació de la paraula per 2.
- **QuadrupleLetter**: Multiplica la puntuació de la lletra per 4.
- **TripleLetter**: Multiplica la puntuació de la lletra per 3.
- **DoubleLetter**: Multiplica la puntuació de la lletra per 2.

3.1.2.5. StandardBoard

Representa un tauler de Scrabble configurat amb la disposició estàndard 15x15. Inicialitza totes les posicions de les caselles de puntuació especial segons les regles oficials de Scrabble.

Funcionalitats principals:

- **Tauler 15x15**: Inicia un tauler de mida 15x15, seguint la disposició estàndard del joc.
- **Caselles premium**: Configura les caselles de puntuació especial, incloent-hi les de **doble paraula**, **triple paraula**, **doble lletra** i **triple lletra** en les posicions establertes per les regles oficials.
- **Còpia**: Implementa la creació d'una còpia del tauler actual, cosa que permet duplicar l'estat del joc.

Aquesta classe estén la classe **Board** i personalitza les caselles de puntuació premium per a la variant oficial del joc.

3.1.2.6. SuperBoard

Representa un tauler de Scrabble de 21x21, utilitzat en modes de joc ampliat i personalitzats. Aquest tauler inclou caselles de puntuació premium addicionals com les de **paraula quàdruple** i **lletra quàdruple**.

Funcionalitats principals:

- **Tauler 21x21**: Inicia un tauler de mida 21x21 per a jocs personalitzats o ampliat.
- **Caselles premium**: Configura caselles amb puntuacions especials, incloent-hi **doble paraula**, **triple paraula**, **quàdruple paraula**, **doble lletra**, **triple lletra** i **quàdruple lletra**, en les posicions específiques per a aquesta variant de tauler.
- **Còpia**: Implementa la creació d'una còpia del tauler actual, que permet duplicar l'estat del joc.

Aquesta classe estén la classe **Board** i personalitza les caselles de puntuació premium per a una variant més gran del joc.

3.1.2.7. CrossChecks

La classe CrossChecks encapsula la lògica de verificació creuada per validar moviments en funció del context del tauler. Estableix les restriccions de col·locació de peces segons les paraules existents, separant les restriccions verticals i horitzontals. Serveix com a infraestructura comuna per adaptar-se a diferents jocs amb diferents conjunts de lletres, mantenint la responsabilitat clara i extensible.

3.1.2.8. CatalanCrossChecks

La classe CatalanCrossChecks és una implementació concreta de les comprovacions creuades per a la versió catalana del joc. Defineix el conjunt complet de lletres vàlides, incloent-hi les formes especials com la "Ç", "L·L" i "NY", i proporciona la lògica necessària per saber si una lletra es pot col·locar en una posició concreta del tauler. És a dir, adapta el comportament genèric de CrossChecks al context lingüístic del català, assegurant la validesa dels moviments en funció de les regles pròpies de l'idioma.

3.1.2.9. EnglishCrossChecks

Aquesta classe, EnglishCrossChecks, adapta el sistema de comprovació de lletres per a partides en anglès. Defineix les 26 lletres estàndard de l'alfabet anglès i utilitza una lògica directa per verificar si una lletra es pot col·locar en una casella, basant-se en les restriccions verticals i horitzontals. Com que l'anglès no té peces especials com el català, la implementació és més senzilla i eficient. Serveix essencialment per validar moviments segons la configuració del tauler i les paraules ja col·locades.

3.1.2.10. SpanishCrossChecks

Aquesta classe SpanishCrossChecks adapta el sistema de validació de lletres del joc per a l'idioma castellà. Té en compte tant les 26 lletres bàsiques com les peces especials pròpies del castellà com "Ñ", "RR", "LL" i "CH", assignant-los posicions específiques. Permet comprovar de manera eficient si una lletra concreta es pot col·locar en una posició del tauler segons les restriccions de les paraules ja col·locades, tant en vertical com en horitzontal. S'assegura així que només es facin jugades vàlides segons el diccionari i la disposició del tauler.

3.1.2.11. DAWG

Aquesta classe DAWG (Directed Acyclic Word Graph⁵) representa l'estructura central per emmagatzemar eficientment paraules d'un diccionari. Serveix com a base per a validacions ràpides de paraules, útil per evitar duplicats i optimitzar la cerca. Manté un node arrel (root) i un conjunt de nodes únics, indexats pel seu hash. Permet accedir ràpidament a qualsevol node, afegir-ne de nous o eliminar-ne, cosa que és útil durant la construcció del graf o quan es fa optimització de memòria.

⁵ Daciuk, J., Watson, B. W., Mihov, S., & Watson, R. E. (2000, 6 juliol). Incremental Construction of Minimal Acyclic Finite-State Automata.

3.1.2.12. Node

Aquesta classe Node representa cada punt del graf de paraules (DAWG). Cada node guarda una lletra (character), una referència al seu pare, la seva profunditat en el graf, si marca el final d'una paraula (isEndOfWord) i els seus successors (els nodes als quals pot avançar). Es calcula un hashCode únic per cada configuració de node, cosa clau per identificar i reutilitzar subgrafos iguals i així optimitzar memòria. És essencial per construir un DAWG compacte i eficient.

3.1.2.13. PlayerDoesNotHavePieceException

Aquesta classe defineix una excepció personalitzada que es llança quan un jugador intenta fer servir una peça que no té a la mà. És útil per controlar errors de lògica en les jugades del jugador.

3.1.2.14. ScrabbleException

La classe ScrabbleException és la classe base abstracta per a totes les excepcions específiques del joc Scrabble. Estén la classe RuntimeException, i això permet que siguin excepcions no comprovades, les quals es poden llençar durant el joc quan es produeixen violacions de regles o accions invàlides.

Aquest enfocament permet definir excepcions específiques per a diversos errors en el joc, com per exemple, jugades invàlides o intent de moure fitxes fora de límits.

3.1.2.15. Leaderboard

Aquesta classe representa un rànquing de puntuacions (Leaderboard) i serveix per emmagatzemar una llista de puntuacions (Score). Permet afegir noves puntuacions i recuperar-les en forma d'array, cosa útil per comunicar-se amb altres parts del sistema com els controladors. És una estructura senzilla que encapsula la llista i centralitza la seva gestió.

3.1.2.16. Score

Aquesta classe Score és un record de Java que emmagatzema informació sobre una partida de Scrabble. Conté tres propietats:

- **scoreValue**: la puntuació total aconseguida pel jugador.
- **isWinner**: un valor booleà que indica si el jugador va guanyar la partida.
- **playerName**: el nom del jugador.

És una manera eficient de representar la informació rellevant d'una partida per després poder gestionar-la en un rànquing.

3.1.2.17. Bag

La classe Bag gestiona una col·lecció de peces per al joc de Scrabble. Permet afegir peces a la bossa, comprovar si està buida, obtenir el nombre de peces que queden i retirar peces per l'índex. És bàsicament una estructura per emmagatzemar i manipular les peces durant el joc.

3.1.2.18. Piece

La classe Piece representa una peça del joc de Scrabble, amb una lletra o dígraf, un valor i la possibilitat de ser una peça en blanc. Ofereix constructors per crear peces amb una lletra i valor fixos, així com amb l'estat en blanc. També proporciona mètodes per obtenir la lletra, el valor i l'estat en blanc de la peça, així com per establir una nova lletra per a les peces en blanc.

3.1.2.19. GameProperties

La classe GameProperties és un record que emmagatzema la informació relacionada amb una partida de Scrabble. Inclou els següents atributs:

- **language:** La llengua en què es juga la partida.
- **boardType:** El tipus de tauler seleccionat per al joc.
- **players:** Una llista de noms dels jugadors humans.
- **cpus:** Una llista de noms dels jugadors controlats per la CPU.

S'utilitza principalment per centralitzar les propietats essencials per a la creació d'una partida.

3.1.2.20. Language

La classe Language és un enumerat que defineix les tres llengües possibles per a una partida de Scrabble:

- **English:** L'idioma anglès.
- **Catalan:** L'idioma català.
- **Spanish:** L'idioma castellà.

Els enumerats són útils per restringir un conjunt de valors possibles i en aquest cas, s'utilitzen per establir el llenguatge de la partida.

3.1.2.21. Movement

Aquest registre emmagatzema la configuració d'una partida de Scrabble. La seva finalitat és contenir les propietats bàsiques de la partida, com ara el **llenguatge**, el **tipus de tauler**, la llista de **jugadors humans** i la llista de **jugadors de la CPU**.

Propietats:

- **language:** L'idioma que s'utilitzarà per al joc.
- **boardType:** El tipus de tauler que s'utilitzarà a la partida, com el tauler estàndard o algun tipus de tauler personalitzat.
- **players:** Una llista amb els noms dels jugadors humans que participaran en la partida.
- **cpus:** Una llista amb els noms dels jugadors controlats per la CPU, si n'hi ha.

3.1.2.22. Player

La classe Player representa un jugador de Scrabble amb els seus atributs principals: nom, puntuació, tipus (humà o CPU), i una mà de peces. Inclou mètodes per obtenir el nom, la puntuació, i la mà de peces, així com per afegir i eliminar peces. Si el jugador intenta eliminar una peça que no té, es llança una excepció. La classe també gestiona l'ús de peces en blanc.

3.2.2. Descripció

3.2.2.1. DrawActionMaker

Aquesta classe DrawActionMaker s'encarrega de gestionar l'intercanvi de fitxes entre la mà del jugador i la bossa de fitxes. Si el jugador vol canviar algunes fitxes:

- Es comprova que n'hi hagi prou a la bossa.
- S'intercanvien amb fitxes noves (usant PieceDrawer).
- S'actualitza la visualització de la mà (IHandDisplay).

És una capa de coordinació entre dades (Bag, Player) i accions (PieceDrawer, actualització de mà).

3.2.2.2. PlaceActionMaker

Gestiona totes les accions necessàries per col·locar una paraula al tauler de Scrabble. S'encarrega de validar que el moviment sigui vàlid (dins dels límits del tauler, que la paraula existeixi, que estigui connectada a altres paraules o col·locada al centre si és el primer torn), de col·locar la paraula, i d'actualitzar l'estat del joc.

3.2.2.3. SkipActionMaker

La classe SkipActionMaker serveix per encapsular l'acció de passar torn en una partida de Scrabble. És una classe simple però útil per separar responsabilitats i mantenir el codi modular.

3.2.2.4. AI

Classe abstracta que implementa la lògica de la intel·ligència artificial per trobar la millor jugada possible al Scrabble. Explora totes les paraules que es poden formar des dels punts d'ancoratge del tauler utilitzant les peces disponibles i calcula quina combinació dona més punts. Permet tractar casos especials de cada idioma mitjançant mètodes abstractes.

3.2.2.5. CatalanAI

Classe que implementa la lògica específica de la intel·ligència artificial per al català. Estén la classe abstracta AI i afegeix el tractament de lletres compostes com NY i L·L, assegurant que es considerin com a peces especials tant en la generació de paraules com en la validació de les jugades.

3.2.2.6. EnglishAI

Classe que implementa la lògica específica de la intel·ligència artificial per a l'anglès. Estén la classe AI i aplica les regles ortogràfiques i lèxiques pròpies de la llengua anglesa, sense requerir un tractament especial de lletres compostes.

3.2.2.7. SpanishAI

Classe que implementa la lògica específica de la intel·ligència artificial per al castellà. Estén la classe AI i inclou el tractament de lletres compostes com LL i CH, permetent reconèixer-les com a unitats lèxiques vàlides durant la generació i avaluació de paraules.

3.2.2.8. IBoard

Aquesta interfície, anomenada IBoard, serveix per comunicar el domini amb la presentació. Defineix un únic mètode, `updateBoard`, que s'utilitza per actualitzar visualment el tauler de joc a la interfície d'usuari quan es produeixen canvis en l'estat del joc.

3.2.2.9. PointCalculator

La classe `PointCalculator` és responsable de calcular els punts associats amb una jugada en el tauler de Scrabble. La classe té en compte factors com el valor de les peces individuals, els multiplicadors de paraules, les bonificacions i les interaccions amb les peces ja col·locades al tauler.

Funcionalitats clau:

- **Càlcul de punts per peça:** Calcula els punts que cada peça aporta a la jugada, tenint en compte els multiplicadors de les caselles (ex: caselles de lletra doble o triple).
- **Multiplicador de paraules:** Calcula el multiplicador aplicable a la paraula formada per les peces col·locades.
- **Jugada de paraules presents:** Calcula els punts de les paraules que ja estaven presents al tauler i que es veuen afectades per la nova jugada.
- **Bonificació de Bingo:** Si la jugada col·loca exactament 7 peces (un "Bingo"), s'assignen 50 punts addicionals.
- **Punts de la paraula ja existent:** Afegeix els punts de les paraules que es poden formar amb les peces noves col·locades al tauler.

La classe integra aquests mètodes per determinar el total de punts per una jugada donada, tenint en compte tots els factors possibles de puntuació dins del joc de Scrabble.

3.2.2.10. PresentPiecesWordCompleter

La classe `PresentPiecesWordCompleter` s'encarrega d'identificar les paraules formades per les peces col·locades al tauler de Scrabble, tant les noves com les adjacents a les peces existents. A través de la detecció de la direcció de les peces (horitzontal o vertical), recupera les paraules completes formades i les afegeix a la llista de paraules. També pot completar les paraules adjacents quan es col·loquen noves peces, garantint així la coherència del joc.

3.2.2.11. WordGetter

La classe `WordGetter` s'encarrega d'identificar les peces que formen una nova paraula al tauler de Scrabble després de col·locar-ne de noves. Aquesta classe gestiona casos on les noves peces expandeixen paraules existents o formen noves paraules. A través de la seva funció principal, recorre les posicions de les peces noves i recupera les peces contínues que formen una paraula, sigui horitzontalment o verticalment. Si es detecten cel·les amb peces existents, també s'afegeixen a la llista, proporcionant així una visió completa de la nova paraula formada.

3.2.2.12. WordPlacer

La classe WordPlacer és responsable de col·locar paraules (peces) al tauler de Scrabble i actualitzar les puntuacions del jugador. Després de col·locar les peces, actualitza la vista per reflectir el nou estat del tauler. La funció principal accepta un conjunt de peces, les posicions d'inici i la direcció (horitzontal o vertical) per col·locar les peces al tauler, calcula la puntuació amb la classe PointCalculator, i actualitza el marcador del jugador. Els mètodes auxiliars placeWordVertical i placeWordHorizontal s'encarreguen de col·locar les peces en les direccions corresponents.

3.2.2.13. CrossCheckUpdater

La classe CrossCheckUpdater s'encarrega de gestionar les comprovacions creuades per validar si les noves peces col·locades en el tauler formen paraules vàlides. Aquesta classe interactua amb el tauler de Scrabble per analitzar les interaccions de les paraules noves amb les existents, tenint en compte la direcció (horitzontal o vertical) i la validació de les paraules mitjançant un diccionari (DAWG).

A grans trets:

1. **Comprovacions Creuades Horitzontals i Verticals:** Per a cada moviment, la classe comprova les caselles a les vores de la paraula col·locada (al principi i al final) per verificar si les lletres col·locades en aquestes posicions poden formar paraules vàlides amb la interacció d'altres lletres al voltant.
2. **Interacció amb DAWG:** Utilitza l'arbre DAWG (Directed Acyclic Word Graph) per comprovar si les paraules formades són vàlides, des del principi fins al final de les noves paraules formades a través de les peces col·locades.
3. **Validació per Idioma:** La classe utilitza un PiecesConverter per garantir que les peces s'ajustin al conjunt de lletres permès per a cada idioma (per exemple, català, espanyol, anglès).
4. **Comprovació de Paraules i Encreuaments:** Per cada casella buida al voltant de la paraula nova, la classe comprova si la nova lletra formada pot formar una paraula vàlida en la direcció horitzontal o vertical. Si no és vàlida, es marca la casella com a "no vàlida" per a aquesta lletra.

En resum, aquesta classe gestiona la lògica de comprovació creuada per assegurar que les noves paraules creades siguin vàlides, tant per a les paraules existents com per a les noves peces col·locades, i això tenint en compte l'idioma i les regles específiques de Scrabble.

3.2.2.14. WordAdder

La classe WordAdder s'encarrega d'afegir paraules al DAWG de manera eficient. Utilitza la paraula anterior afegida per estalviar espai, afegint només els nodes nous necessaris per a la paraula actual.

Funcionalitat:

1. **Afegir Paraules:** Afegeix una nova paraula al DAWG, aprofitant el prefix comú amb la paraula anterior per evitar duplicats de nodes.
2. **Cerca de Prefix Comú:** Determina on comença a diferir la nova paraula de l'anterior per afegir només els nodes restants.
3. **Actualització de Nodes:** Si és necessari, crea nous nodes i els afegeix al DAWG. També marca l'últim node de la paraula com a "final de paraula".

En resum, aquesta classe optimitza l'afegiment de paraules al DAWG aprofitant les paraules ja afegides i millorant l'eficiència.

3.2.2.15. WordValidator

La classe WordValidator s'encarrega de verificar si una paraula és vàlida segons el diccionari representat pel DAWG. La seva funció principal és validar si una paraula existeix al conjunt de paraules emmagatzemades en aquest gràfic.

3.2.2.16. InitialMoveNotInCenterException

La classe InitialMoveNotInCenterException és una excepció personalitzada que es llança quan el primer moviment d'un jugador no es realitza a la casella central del tauler de Scrabble. Aquesta excepció garanteix que la primera paraula jugada a la partida comenci a la casella central, mantenint les regles del joc.

3.2.2.17. MovementOutsideOfBoardException

La classe MovementOutsideOfBoardException és una excepció personalitzada que es llança quan un jugador intenta realitzar un moviment fora dels límits del tauler de Scrabble. Aquesta excepció assegura que els jugadors no puguin col·locar fitxes fora de l'àrea jugable del tauler, respectant les regles del joc.

3.2.2.18. NotEnoughPiecesInBagException

La classe NotEnoughPiecesInBagException és una excepció personalitzada que s'usa quan no hi ha prou fitxes disponibles a la bossa per completar un moviment. Aquesta excepció garanteix que els jugadors no puguin fer un moviment que requereixi més fitxes de les que queden a la bossa.

3.2.2.19. NotEnoughPiecesInHandException

La classe NotEnoughPiecesInHandException és una excepció personalitzada que es llança quan un jugador no té prou fitxes a la seva mà per fer un moviment. Això assegura que la lògica del joc impedeixi moviments invàlids a causa de la falta de fitxes disponibles per part del jugador.

3.2.2.20. WordDoesNotExistException

La classe `WordDoesNotExistException` és una excepció llançada per indicar que una paraula no existeix al diccionari durant una jugada. Aquesta excepció es llança quan un jugador intenta col·locar una paraula al tauler que no és vàlida segons el diccionari de paraules acceptades. Això garanteix que només es puguin utilitzar paraules vàlides durant el joc.

3.2.2.21. WordNotConnectedToOtherWordsException

La classe `WordNotConnectedToOtherWordsException` és una excepció llançada per indicar que una paraula no està connectada a cap altra paraula al tauler. Aquesta excepció es produeix quan un jugador intenta col·locar una paraula al tauler que no està connectada amb les paraules prèviament col·locades, la qual cosa viola la regla que estableix que totes les noves paraules han de connectar-se amb alguna altra paraula existent.

3.2.2.22. GameStepper

La classe `GameStepper` s'encarrega de gestionar el procés del joc, verificant si el joc ha acabat a través de la classe `Turn`. Quan el joc finalitza, actualitza la taula de classificació (Leaderboard) amb els punts dels jugadors i marca qui ha guanyat en funció del jugador amb el màxim de punts.

3.2.2.23. IWinScreen

La interfície `IWinScreen` serveix per establir una comunicació entre la capa de domini i la capa de presentació del joc. Concretament, permet que la capa de domini notifiqui la capa de presentació quan un jugador guanya la partida, activant una acció o actualitzant la vista.

3.2.2.24. GamesPlayedLeaderboard

La classe `GamesPlayedLeaderboard` gestiona la classificació dels jugadors segons el nombre de partides que han jugat. El seu rol principal és processar les puntuacions (un conjunt de `Score`) i generar un llistat dels jugadors ordenats per la quantitat de partides jugades, de manera que els jugadors amb més partides es mostrin primer.

3.2.2.25. GamesWinsPair

La classe `GamesWinsPair` és una classe auxiliar dissenyada per calcular la taxa de victòries d'un jugador. Gestiona el nombre total de partides jugades i el nombre de victòries d'un jugador, i ofereix funcionalitats per actualitzar aquests valors a mesura que es juguen més partides, així com calcular la taxa de victòries.

3.2.2.26. GamesWonLeaderboard

La classe `GamesWonLeaderboard` és responsable de gestionar una taula de classificació basada en el nombre de victòries que ha aconseguit cada jugador en les partides jugades. Ordena els jugadors segons les victòries i retorna una llista d'objectes `PlayerValuePair` que contenen el nom del jugador i el nombre de victòries.

3.2.2.27. MaxScoreLeaderboard

La classe MaxScoreLeaderboard és una classe encarregada de generar una taula de classificació basada en el màxim puntatge obtingut per cada jugador en les partides jugades. Ordena els jugadors segons el seu màxim puntatge i retorna una llista d'objectes PlayerValuePair amb el nom del jugador i el seu màxim puntatge.

3.2.2.28. PlayerValuePair

La classe PlayerValuePair és una classe auxiliar que emmagatzema el nom d'un jugador juntament amb un valor associat a aquest jugador. Aquesta classe és utilitzada en diverses parts del sistema, especialment per gestionar la taula de classificació (leaderboard), per emmagatzemar el nom dels jugadors i els valors que els associen (com el nombre de jocs jugats, el nombre de jocs guanyats o el màxim puntatge obtingut).

3.2.2.29. TotalScoreLeaderboard

La classe TotalScoreLeaderboard és una classe controladora que s'encarrega de generar una taula de classificació ordenada en funció del total de punts aconseguits per cada jugador en diverses rondes o partides. Aquesta classe s'orienta a l'ús en un sistema de classificació per puntuació total i actua sobre una col·lecció de resultats de puntuacions (representats per objectes Score).

3.2.2.30. WinRateLeaderboard

La classe WinRateLeaderboard és una classe controladora que genera una taula de classificació basada en el percentatge de victòries (win rate) aconseguides per cada jugador. A partir d'un conjunt de resultats de partides, calcula la taxa de victòries de cada jugador i ordena els jugadors en funció d'aquest valor. La classe utilitza una estructura de dades addicional GamesWinsPair per emmagatzemar el nombre de partides jugades i guanyades per cada jugador.

3.2.2.31. MovementBoundsChecker

La classe MovementBoundsChecker s'utilitza per comprovar si un moviment (conjunt de lletres) dins d'un joc de Scrabble es troba dins dels límits del tauler de joc. Aquesta classe verifica que les coordenades de les caselles que contindran la paraula estiguin dins de les fronteres del tauler i siguin vàlides per a la seva col·locació.

3.2.2.32. MovementCleaner

La classe MovementCleaner s'utilitza per gestionar el procés de neteja durant la col·locació d'una nova paraula en un joc de Scrabble. L'objectiu és assegurar-se que les peces de la nova paraula es col·loquin correctament en el tauler i que es faci només en aquelles caselles que estiguin buides o coincidint amb peces que ja hi siguin col·locades, evitant superposicions incorrectes.

3.2.2.33. BagFiller

La classe BagFiller s'utilitza per omplir una bossa de joc amb peces específiques de Scrabble, segons la configuració de llenguatge proporcionada. Aquesta classe és crucial per inicialitzar el joc amb les peces correctes i assegurar que el conjunt de peces sigui adequat per jugar.

3.2.2.34. CatalanPiecesConverter

La classe CatalanPiecesConverter és una implementació de la classe abstracta PiecesConverter i s'utilitza per convertir una cadena de lletres (una paraula) en peces de Scrabble, específicament per a la llengua catalana. Aquesta classe és útil per gestionar casos especials de combinacions de lletres, com "L·L" i "NY", que són típics en català.

3.2.2.35. EnglishPiecesConverter

La classe EnglishPiecesConverter és una implementació de la classe abstracta PiecesConverter. En aquest cas, la classe no implementa cap funcionalitat especial, ja que l'anglès no compta amb un set de caràcters especial, sinó que s'utilitza únicament per ajudar a entendre millor l'objectiu d'aquesta classe.

3.2.2.36. HandFiller

La classe HandFiller és responsable de distribuir les peces a cada jugador al començament del joc de Scrabble. Quan s'inicia el joc, cada jugador ha de rebre 7 peces aleatòries del sac de peces disponible. La classe utilitza un generador de nombres aleatoris, implementat a través de la interfície IRand, per seleccionar les peces de manera aleatòria.

El procés de distribució es fa de la següent manera:

- **Verificació de disponibilitat:** Primer, la classe comprova que hi ha prou peces al sac per a tots els jugadors (7 peces per jugador).
- **Distribució de peces:** Per cada jugador, se li assignen 7 peces aleatòries seleccionades del sac utilitzant el mètode draw del sac (Bag).

Finalment, cada jugador rep les seves 7 peces de manera independent. Aquesta classe és fonamental per assegurar-se que cada jugador comenci amb les peces adequades i que el joc es pugui desenvolupar de manera equilibrada.

3.2.2.37. IFileReader

Aquesta interfície IFileReader defineix un contracte per a qualsevol classe que vulgui encarregar-se de llegir fitxers (o contingut relacionat) segons l'idioma (Language) indicat i retornar el contingut en un String.

3.2.2.38. PieceDrawer

La classe PieceDrawer gestiona l'intercanvi de peces entre la mà d'un jugador i la bossa de joc, permetent que el jugador retorni peces no desitjades i en tregui de noves de la bossa.

3.2.2.39. PieceGenerator

La classe PieceGenerator és responsable de generar objectes Piece a partir d'un fitxer que conté informació sobre les peces del joc. Llegeix el contingut del fitxer línia per línia, on cada línia conté informació sobre una peça (caràcter, valor i quantitat). Per cada línia, crea una peça i emmagatzema la quantitat corresponent a un array de parelles (Pair<Piece, Integer>), on Piece representa la peça i el nombre d'exemplars de cada una.

Aquest procés permet generar de manera eficient totes les peces que es faran servir en el joc.

3.2.2.40. **PiecesConverter**

La classe abstracta `PiecesConverter` gestiona la conversió d'una cadena de caràcters a un conjunt de peces de Scrabble. La classe maneja únicament les peces amb caràcters presents a l'alfabet anglès.

3.2.2.41. **PiecesConverterFactory**

La classe `PiecesConverterFactory` s'encarrega de crear instàncies de la classe `PiecesConverter` corresponents a un idioma determinat, facilitant així la conversió i gestió de les peces del joc de Scrabble segons la llengua seleccionada.

3.2.2.42. **PiecesInHandGetter**

La classe `PiecesInHandGetter` és responsable de gestionar les peces que un jugador pot tenir a la seva mà i de verificar si les peces actuals són vàlides per a un moviment. També permet intercanviar peces entre la mà del jugador i la bossa de peces.

3.2.2.43. **PiecesInHandVerifier**

La classe `PiecesInHandVerifier` és responsable de verificar si un jugador té les peces necessàries a la seva mà per formar una paraula determinada. Aquesta classe fa servir un conversor de peces (com `PiecesConverter`) per convertir la paraula en un conjunt de peces, i després comprova si aquestes peces es troben a la mà del jugador.

3.2.2.44. **SpanishPiecesConverter**

La classe `SpanishPiecesConverter` és una especialització de la classe `PiecesConverter` dissenyada per gestionar la conversió de paraules a peces específiques per a la versió en espanyol de Scrabble. Aquesta classe s'encarrega de gestionar casos especials com les combinacions de lletres "RR", "LL" i "CH", que són considerades com una sola peça en el joc de Scrabble en espanyol.

3.2.2.45. **Endgame**

La classe `Endgame` serveix per determinar si una partida de Scrabble ha acabat. Aquesta classe es basa en dos factors per decidir si el joc ha arribat al final:

- **Límits de passar el torn:** Si tots els jugadors han saltat tres vegades consecutives, es considera que el joc ha acabat.
- **Mà buida:** Si un jugador s'ha quedat sense peces a la mà, es considera que el joc ha acabat.

3.2.2.46. IGamePlayer

La interfície IGamePlayer defineix els mètodes que gestiona un jugador durant el seu torn en el joc. Conté tres mètodes principals:

- **startTurn**: S'executa quan comença el torn del jugador, indicant que el jugador pot realitzar accions, com jugar una paraula a tauler.
- **endTurn**: Es crida quan el jugador acaba les seves accions durant el torn, i retorna un resultat encapsulat en un objecte TurnResult, què pot incloure informació sobre els punts obtinguts o la validesa de la jugada.
- **isActive**: Comprova si el jugador està actualment jugant el seu torn. Retorna true si el jugador està actiu i jugant, o false si ha acabat el seu torn o encara no ha començat.

Aquesta interfície permet gestionar de manera ordenada els torns de cada jugador en el joc.

3.2.2.47. Turn

La classe Turn és responsable de gestionar el flux dels torns en una partida de Scrabble. El seu objectiu és controlar quin jugador comença i acaba cada torn, així com realitzar un seguiment de les condicions de finalització de la partida a través de la classe Endgame.

3.2.2.48. TurnResult

L'enumeració TurnResult defineix els possibles resultats d'un torn en el joc Scrabble. Aquesta classe enum especifica tres possibles resultats que un jugador pot obtenir en el seu torn:

- **Skip**: Utilitzat per representar un torn en què el jugador ha passat.
- **Place**: Es fa servir quan el jugador ha realitzat una jugada vàlida (posar una paraula en el tauler).
- **Draw**: Indica que el jugador ha decidit intercanviar peces en lloc de jugar al tauler.

3.2.2.49. AnchorUpdater

La classe AnchorUpdater gestiona l'actualització dels "ancoratges" (cel·les de l'escenari de Scrabble que marquen posicions vàlides per col·locar peces noves) després que un jugador faci un moviment. Aquesta classe és responsable d'actualitzar les cel·les de l'escenari segons la direcció i la posició de les peces que es col·loquen.

3.3. Capa de presentació

3.3.1. Diagrama

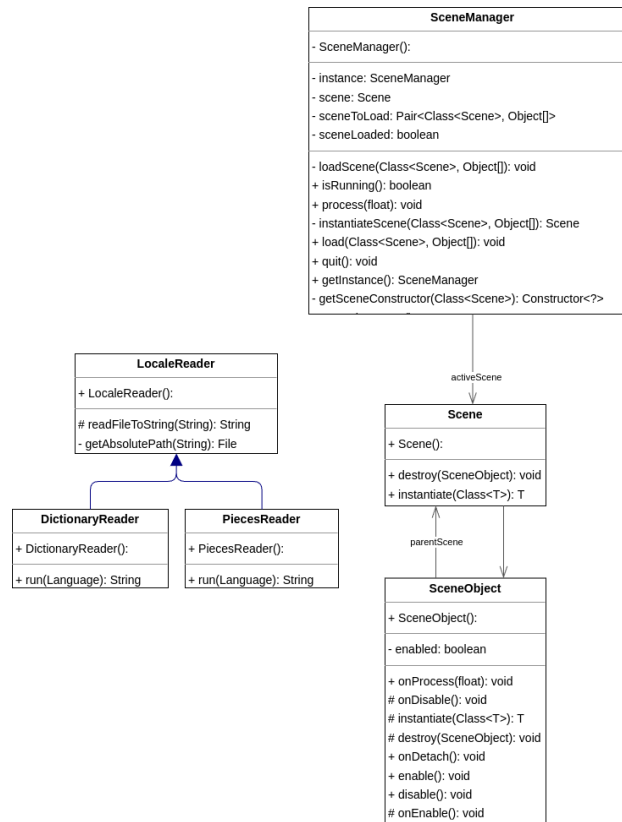


Figura 5: Imatge del diagrama de la capa de presentació

En aquest diagrama, únicament representem els objectes del mòdul d'escenes i de localització. No hi representem els mòduls de les vistes, ja que tal com hem mencionat a l'inici del resum, l'objectiu d'aquesta entrega no és tenir una vista definitiva per al joc. La vista que hi ha present és temporal i únicament està present per tal de testar que la comunicació entre capes funcioni correctament, creiem que no és important veure l'estructura de classes que segueix actualment. Tot i això, en el següent apartat hem explicat breument el funcionament de les classes temporals.

3.3.2. Descripció

3.3.2.1. DictionaryReader

La classe DictionaryReader és responsable de llegir fitxers de diccionari en diferents idiomes, en funció de la selecció de l'idioma (localització). Aquesta classe hereta de LocaleReader i implementa la interfície IFileReader. El seu mètode run rep un paràmetre locale que especifica l'idioma del diccionari desitjat i retorna el contingut del fitxer de diccionari corresponent en forma de cadena de text.

Els idiomes suportats són:

- **Català** (llegeix catalan.txt).
- **Castellà** (llegeix castellano.txt).
- **Anglès** (llegeix english.txt).

El mètode utilitza un switch per determinar quin fitxer llegir segons l'idioma seleccionat. Això facilita la càrrega dinàmica de diccionaris segons la preferència de l'usuari.

3.3.2.2. LocaleReader

La classe LocaleReader llegeix fitxers de localització (com diccionaris) utilitzant una ruta dinàmica per localitzar els fitxers en el projecte. Té dos mètodes principals:

- `getAbsolutePath`: Localitza el fitxer dins de l'estructura del projecte.
- `readFileToString`: Llegeix el contingut del fitxer i el retorna com una cadena de text, amb gestió d'errors si hi ha problemes per llegir-lo.

Es fa servir com a base per a altres classes que necessiten llegir fitxers de localització.

3.3.2.3. PiecesReader

La funcionalitat principal de la classe PiecesReader és llegir fitxers que contenen informació sobre les peces de Scrabble (com les lletres i la seva distribució) en diferents idiomes. Aquesta classe permet obtenir el contingut d'aquests fitxers segons l'idioma que s'indiqui, sigui català, castellà o anglès. D'aquesta manera, el joc pot gestionar les peces de manera específica segons la llengua triada pel jugador, facilitant la personalització i adaptació a diversos mercats i idiomes.

3.3.2.4. SceneObjectWithParametrizedConstructorException

La classe SceneObjectWithParametrizedConstructorException és una excepció dissenyada per gestionar escenaris en què s'intenta instanciar un SceneObject amb un constructor que rep paràmetres, cosa que no és compatible amb el disseny del sistema.

3.3.2.5. Scene

La classe Scene és una classe base dissenyada per gestionar els objectes dins d'una escena en un joc o aplicació. Aquesta classe permet actualitzar els objectes que hi ha dins de l'escena de manera contínua, desactivar-los i eliminar-los quan ja no són necessaris. El seu objectiu és proporcionar una manera organitzada de manipular els objectes que formen part d'una escena dinàmica.

El mètode principal de la classe és `onProcess`, que actualitza els objectes presents dins de l'escena en funció del temps transcorregut. Quan l'escena es desactiva, el mètode `onDetach` s'encarrega de deshabilitar tots els objectes i desvincular-los de l'escena. Per afegir nous objectes, el mètode `instantiate` permet instanciar un objecte a partir d'una classe proporcionada, però llançarà una excepció si la classe té un constructor amb paràmetres, ja que aquesta classe no els suporta.

La classe també disposa de la capacitat de destruir objectes de l'escena mitjançant el mètode `destroy`, que elimina un objecte de la llista d'objectes de l'escena i el deshabilita. A més, es fa un ús de reflexió per identificar el constructor més adequat de cada classe d'objectes i assegurar-se que els objectes es creen sense problemes.

3.3.2.6. SceneManager

La classe `SceneManager` és responsable de gestionar totes les escenes presents en el joc i de gestionar la transició entre elles. El seu principal objectiu és carregar, intercanviar i destruir escenes de manera eficient, assegurant-se que es carreguin les dependències adequades quan es canvia d'escena.

El patró Singleton s'utilitza per garantir que només existeix una instància de `SceneManager` durant l'execució del joc. La classe ofereix la funcionalitat de carregar una nova escena amb el mètode `load`, el qual gestiona el canvi entre escenes actives i noves. Si una escena ja està carregada, es destruirà abans de carregar la nova. Quan es carrega una nova escena, es fa mitjançant reflexió, cercant el constructor més adequat per instanciar la classe d'escena amb les dependències corresponents.

A més, el `SceneManager` també es fa càrrec de la neteja de recursos quan el joc es tanca, amb el mètode `quit`, que desactiva l'escena activa abans de finalitzar. El mètode `process` és cridat contínuament per actualitzar l'escena activa i els seus objectes, i gestiona qualsevol nova escena que hagi estat planificada per carregar-se. Finalment, el mètode `isRunning` permet saber si hi ha una escena activa o no.

3.3.2.7. SceneObject

La classe `SceneObject` és la base per representar qualsevol objecte dins d'una escena. Aquesta classe proporciona funcionalitats bàsiques per habilitar i deshabilitar els objectes de l'escena, així com per gestionar la seva integració amb el sistema de gestió d'escenes. Els objectes de la classe `SceneObject` poden ser instanciats, actualitzats i destruïts segons sigui necessari en el flux del joc.

Cada objecte d'escena està associat a una escena mitjançant el mètode `setScene()`, que permet establir la relació entre l'objecte i l'escena a la qual pertany. L'objecte pot ser habilitat o deshabilitat mitjançant els mètodes `enable()` i `disable()`, respectivament, i aquests mètodes criden els mètodes protegits `onEnable()` i `onDisable()`, que poden ser sobrecarregats per les subclasses per implementar un comportament personalitzat.

Aquesta classe també inclou el mètode `onProcess()`, que s'executa durant l'actualització de l'escena per permetre que l'objecte realitzi alguna acció específica (pot ser sobrecarregat per les subclasses). Quan un objecte es desconnecta de l'escena, es crida el mètode `onDetach()`, que també pot ser sobrecarregat per fer front a la lògica associada a la desconexió.

A més, es proporciona un mecanisme per instanciar nous objectes d'escena i destruir els objectes existents mitjançant els mètodes `stantiate()` i `destroy()`, respectivament. Aquestes accions interactuen directament amb el sistema de gestió d'escenes, facilitant la creació i destrucció d'objectes dins d'una escena.

3.3.2.8. HandView

La classe `HandView` implementa la interfície `IHandDisplay` i es responsabilitza de mostrar la mà actual d'un jugador al terminal. Quan el mètode `updateHand(Piece[] hand)` és cridat, imprimeix les peces de la mà del jugador a la consola.

3.3.2.9. MovementParser

La classe `MovementParser` s'encarrega de parsejar una entrada en format de cadena, com per exemple "PARAULA AB", i convertir-la en un objecte `Movement`. Aquest objecte conté la paraula jugada, la seva posició (en format de coordenades) i la direcció del moviment (horitzontal o vertical). El parser interpreta la paraula, la coordenada associada i la direcció per generar un moviment vàlid dins del joc Scrabble.

3.3.2.10. AIPlayerObject

La classe `AIPlayerObject` representa un jugador controlat per l'AI en el joc, amb funcionalitats per configurar l'AI que decidirà els seus moviments i el processament d'aquests. Quan arriba el torn de l'AI s'inicia el seu torn i es crida al mètode `run` per trobar el millor moviment possible i executar-lo. En cas de no trobar cap moviment en $\frac{1}{4}$ dels casos passarà de torn. Al $\frac{3}{4}$ restant descartarà aproximadament el $\frac{1}{2}$ de la seva mà i agafarà peces noves.

3.3.2.11. HumanPlayerObject

La classe `HumanPlayerObject` representa un jugador humà en el joc, amb funcionalitats per iniciar el torn, seleccionar i processar moviments. Quan el jugador inicia el torn, se li mostren les instruccions de joc i es restaura l'estat de selecció del moviment. Durant el torn, el jugador pot seleccionar una acció (col·locar una paraula, agafar peces o passar el torn) mitjançant comandes específiques. Si el jugador escull col·locar una paraula, la classe utilitza el `MovementParser` per parsejar la comanda escrita i generar un objecte `Movement` que es col·locarà al tauler, actualitzant la puntuació del jugador.

El codi controla les interaccions del jugador amb la terminal, gestionant els moviments i les opcions disponibles segons la comanda introduïda, i imprimeix els resultats o les instruccions pertinents.

3.3.2.12. PlayerObject

La classe PlayerObject és una classe base que representa un jugador en el joc de Scrabble. Implementa la interfície IGamePlayer i hereta de SceneObject, cosa que li permet interactuar amb el sistema de scenes. Aquesta classe maneja la configuració del jugador, incloent-hi la seva informació (a través de l'objecte Player) i la lògica associada als moviments del jugador (com col·locar lletres, agafar peces o saltar el torn).

Els mètodes principals permeten començar el torn del jugador, imprimir les peces que té disponibles, i realitzar accions com col·locar una peça (placePiece), agafar una nova peça (drawPiece), o saltar el torn (skipTurn). Quan el torn acaba, es retorna el resultat del torn a través del mètode endTurn.

Aquesta classe es configura amb un PlaceActionMaker per manejar la col·locació de peces al tauler, i permet el control del flux del joc, gestionant el canvi entre els estats del jugador i les accions disponibles durant el seu torn.

3.3.2.13. GameScene

La classe GameScene és responsable de la configuració i inicialització de tots els elements necessaris per al desenvolupament d'una partida de Scrabble. La seva funció principal és resoldre les dependències que intervenen en una partida, com ara la creació i configuració dels jugadors, el tauler, les peces i altres components del joc, així com la configuració dels jugadors per gestionar les seves accions durant la partida.

El procés comença amb la creació del tauler de joc a partir de les propietats proporcionades. En funció de l'idioma seleccionat (Català, Espanyol o Anglès), es configura un conjunt de components específics, com el sistema de validació de paraules (CrossChecks), el conjunt de paraules (DAWG), i el lector de diccionaris i peces. També es crea el Leaderboard per gestionar les puntuacions, així com altres objectes necessaris per al càlcul dels punts i la validació de les jugades.

Els jugadors es creen a partir de les dades proporcionades per les propietats del joc, diferenciant-se entre jugadors humans i jugadors automàtics (AI). Per a cada jugador, es genera un objecte específic (AIPlayerObject o HumanPlayerObject) i es configura per tal de permetre-li interactuar amb el sistema de joc. A més, es configura l'acció de col·locar peces (PlaceActionMaker), el sistema de validació dels moviments, la gestió del tauler, el càlcul de punts, i altres aspectes del flux de joc.

El mètode configurePlayers assigna a cada jugador un conjunt d'objectes necessaris per poder realitzar les seves accions en el joc, com ara col·locar paraules, agafar peces i gestionar les jugades. Quan tot està configurat, els jugadors estan preparats per començar el seu torn.

Aquesta classe gestiona el flux general de la partida, assegurant-se que tots els components del joc estan correctament inicialitzats i connectats, i també permet que els jugadors realitzin les seves jugades en funció de les seves accions.

3.3.2.14. MenuScene

La classe MenuScene representa una escena en el joc Scrabble, en què es presenten opcions per començar una nova partida, carregar una partida anterior, o sortir del joc. En aquest cas, la classe simplement crea una instància de la classe GameCreator, que serà la responsable de gestionar el procés de creació de la partida (com l'elecció del tauler, la configuració dels jugadors i l'idioma).

3.3.2.15. Reader

La classe Reader és una utilitat per llegir línies de text de manera asíncrona en un entorn de consola. Es tracta d'un patró de disseny conegut com a "singleton", que assegura que només hi hagi una instància d'aquesta classe durant l'execució de l'aplicació, ja que a java només una classe pot estar llegint de la terminal. La seva principal funcionalitat és gestionar la lectura de línies d'entrada a través d'un fil (thread) dedicat que utilitza un BlockingQueue per emmagatzemar les línies llegides, permetent una lectura no bloquejant quan la resta de l'aplicació segueix executant-se.

3.3.2.16. BoardView

La classe BoardView és responsable de representar i actualitzar la visualització del tauler del joc de Scrabble en la terminal. Implementa la interfície IBoard, la qual indica que la classe té la capacitat d'actualitzar la visualització del tauler del joc. A continuació, et dono una explicació detallada del seu funcionament:

3.3.2.17. GameCreator

La classe GameCreator és responsable de gestionar el procés de configuració inicial d'una partida de Scrabble en la terminal. A través d'una sèrie d'interfícies interactives, els usuaris poden seleccionar l'idioma, la mida del tauler i els jugadors que participen en la partida.

3.3.2.18. LanguageSetter

La classe LanguageSetter és responsable de gestionar la selecció de l'idioma per a la partida de Scrabble en la versió de terminal. Permet a l'usuari navegar entre els idiomes disponibles (anglès, català i castellà) i carrega els arxius associats a cada idioma per a les paraules i les lletres.

3.3.2.19. PieceDisplay

La classe PieceDisplay implementa la interfície IPiecePrinter per mostrar les peces del joc Scrabble en la terminal. Aquesta classe és essencial per a la representació visual de les peces del joc durant la partida.

3.3.2.20. PlayerSetter

La classe PlayerSetter és responsable de gestionar els jugadors i els "bots" (jugadors controlats per la màquina) en la creació d'una partida de Scrabble a la terminal. Aquesta classe gestiona les accions dels jugadors, afegint i eliminant jugadors humans i bots, així com canviant el mode entre afegir jugadors o bots. També és responsable de mostrar la informació sobre els jugadors i els bots de manera visual a la terminal.

3.3.2.21. **SizeSetter**

La classe **SizeSetter** és responsable de gestionar la selecció de la mida del tauler per al joc de Scrabble. Aquesta classe permet als usuaris seleccionar entre tres tipus de taulells disponibles: **Junior**, **Standard** i **Super**. L'usuari pot canviar entre aquestes opcions mitjançant les comandes de la terminal.

3.3.2.22. **TileConverter**

La classe **TileConverter** és una classe utilitzada per convertir diferents tipus de dades associades a les fitxes del joc de Scrabble a representacions textuais que poden ser mostrades a la terminal.

3.4. **Paquet d'utilitats**

3.4.1. **Direction**

La classe **Direction** és un enum que representa les dues possibles direccions en què es pot col·locar una paraula en un tauler de Scrabble.

3.4.2. **IRand**

La interfície **IRand** defineix mètodes per generar números aleatoris, la qual cosa pot ser útil per a diversos aspectes del joc, com ara la selecció de fitxes o la seva ubicació al tauler de Scrabble.

3.4.3. **Pair**

La classe **Pair<X, Y>** és una implementació senzilla d'un parell genèric que pot contenir dos elements de tipus potencialment diferent. Aquesta classe permet emmagatzemar i accedir fàcilment a dues dades que es volen vincular entre elles.

3.4.4. **Rand**

La classe **Rand** és una implementació concreta de la interfície **IRand** que utilitza la classe `java.util.Random` per generar números aleatoris.

3.4.5. **Vector2**

La classe **Vector2** representa un vector bidimensional amb coordenades enters (x, y). Aquesta classe proporciona operacions bàsiques sobre vectors, com ara l'addició de vectors i la sobrecàrrega dels mètodes d'igualtat i de representació en cadena.

4. Descripció dels tests

4.1. TestDrawActionMaker

4.1.1. successSwap

Objecte de prova: DrawActionMaker

Altres elements integrats a la prova: Bag, Player, Piece, Turn, IEndScreen, GameStepper, HandView, IRand.

Valors estudiats: S'intercanvia la peça "b" de la mà del jugador per la peça "a" de la bossa.

Efectes estudiats: El jugador ha canviat la peça seleccionada per una de la bossa. Ara el jugador té la peça "a" i a la bossa hi ha la peça "b". La vista de la mà del jugador s'actualitza.

4.1.2. piecesToSwapNull

Objecte de prova: DrawActionMaker

Altres elements integrats a la prova: Bag, Player, Piece, Turn, IEndScreen, GameStepper, HandView, IRand.

Valors estudiats: S'intercanvia una peça del jugador tenint una bossa buida.

Efectes estudiats: Es llança una excepció de tipus IllegalArgumentException.

4.1.3. notEnoughBagPieces

Objecte de prova: DrawActionMaker

Altres elements integrats a la prova: Bag, Player, Piece, Turn, IEndScreen, GameStepper, HandView, IRand.

Valors estudiats: S'intercanvien 2 peces "b" del jugador tenint 1 única peça "a" a la bossa.

Efectes estudiats: Es llança una excepció de tipus NotEnoughPiecesInBagException.

4.1.4. multipleBagPieces

Objecte de prova: DrawActionMaker

Altres elements integrats a la prova: Bag, Player, Piece, Turn, IEndScreen, GameStepper, HandView, IRand.

Valors estudiats: S'intercanvien 2 peces "b" del jugador tenint 4 peces "a" a la bossa.

Efectes estudiats: La vista de la mà del jugador s'actualitza. El jugador té 2 peces "a" i a la bossa hi ha 2 peces "a" i 2 peces "b".

4.2. TestPlaceActionMaker

4.2.1. placePieceSendsUpdateViewCallback

Objecte de prova: *PlaceActionMaker*

Altres elements integrats a la prova: *Board, Bag, DAWG, Player, WordGetter, PointCalculator, WordPlacer, PointCalculator, WordValidator, PiecesInHandGetter, MovementCleaner, PresentPiecesWordCompleter, CrossChecks, CrossCheckUpdater, Turn, GameStepper*

Valors estudiats: Es fa un moviment col·locant la paraula “POTATO” en horitzontal començant per la casella (5, 7)

Efectes estudiats: La vista del tauler rep un *callback* d’actualització

4.2.2. placePieceThrowsExceptionIfWordDoesNotExist

Objecte de prova: *PlaceActionMaker*

Altres elements integrats a la prova: *Board, Bag, DAWG, Player, WordGetter, PointCalculator, WordPlacer, PointCalculator, WordValidator, PiecesInHandGetter, MovementCleaner, PresentPiecesWordCompleter, CrossChecks, CrossCheckUpdater, Turn, GameStepper*

Valors estudiats: Es fa un moviment col·locant la paraula “POTATO” en horitzontal començant per la casella (5, 7), sense que aquesta paraula estigui present al diccionari

Efectes estudiats: Es llança una excepció de tipus *WordDoesNotExistException*

4.2.3. placePieceThrowsExceptionIfMovementIsOutsideOfBoard

Objecte de prova: *PlaceActionMaker*

Altres elements integrats a la prova: *Board, Bag, DAWG, Player, WordGetter, PointCalculator, WordPlacer, PointCalculator, WordValidator, PiecesInHandGetter, MovementCleaner, PresentPiecesWordCompleter, CrossChecks, CrossCheckUpdater, Turn, GameStepper*

Valors estudiats: Es fa un moviment col·locant la paraula “POTATO” en horitzontal començant per la casella (10, 1)

Efectes estudiats: Es llança una excepció de tipus *MovementOutsideOfBoardException*

4.2.4. **placePieceThrowsExceptionIfContainsCombinedWordThatDoesNotExist**

Objecte de prova: *PlaceActionMaker*

Altres elements integrats a la prova: *Board, Bag, DAWG, Player, WordGetter, PointCalculator, WordPlacer, PointCalculator, WordValidator, PiecesInHandGetter, MovementCleaner, PresentPiecesWordCompleter, CrossChecks, CrossCheckUpdater, Turn, GameStepper*

Valors estudiats: Es col·loca la paraula “HOLA” en horitzontal a la casella (7, 7), la paraula “ALL” en vertical a la casella (10, 7) i la paraula “COLD” en horitzontal a la casella (8, 8)

Efectes estudiats: Es llança una excepció de tipus *WordDoesNotExistException*, ja que es formen les noves paraules “OC” i “LL”, les quals no existeixen

4.2.5. **placePiecesSubtractsPiecesFromPlayer**

Objecte de prova: *PlaceActionMaker*

Altres elements integrats a la prova: *Board, Bag, DAWG, Player, WordGetter, PointCalculator, WordPlacer, PointCalculator, WordValidator, PiecesInHandGetter, MovementCleaner, PresentPiecesWordCompleter, CrossChecks, CrossCheckUpdater, Turn, GameStepper*

Valors estudiats: Es col·loca la paraula “HOLA” en horitzontal a la casella (7, 7) i existeix una *Bag* que està buida.

Efectes estudiats: El *Player* deixa de tenir les peces corresponents a aquella paraula.

4.2.6. **placedPiecesInHandGetReplacedByBagPieces**

Objecte de prova: *PlaceActionMaker*

Altres elements integrats a la prova: *Board, Bag, DAWG, Player, WordGetter, PointCalculator, WordPlacer, PointCalculator, WordValidator, PiecesInHandGetter, MovementCleaner, PresentPiecesWordCompleter, CrossChecks, CrossCheckUpdater, Turn, GameStepper*

Valors estudiats: Es col·loca la paraula “HOLA” en horitzontal a la casella (7, 7) i existeix una *Bag* amb les lletres T, E, S i T.

Efectes estudiats: El *Player* deixa de tenir les peces corresponents a aquella paraula i passa a tenir les lletres T, E, S i T

4.2.7. placedPiecesGetPlacedUpdatesBoard

Objecte de prova: *PlaceActionMaker*

Altres elements integrats a la prova: *Board, Bag, DAWG, Player, WordGetter, PointCalculator, WordPlacer, PointCalculator, WordValidator, PiecesInHandGetter, MovementCleaner, PresentPiecesWordCompleter, CrossChecks, CrossCheckUpdater, Turn, GameStepper*

Valors estudiats: Es col·loca la paraula “HOLA” en horitzontal a la casella (7, 7)

Efectes estudiats: Al tauler es poden veure les peces que s’han col·locat

4.2.8. continuousPlacesCorrectlyPlacedOnBoard

Objecte de prova: *PlaceActionMaker*

Altres elements integrats a la prova: *Board, Bag, DAWG, Player, WordGetter, PointCalculator, WordPlacer, PointCalculator, WordValidator, PiecesInHandGetter, MovementCleaner, PresentPiecesWordCompleter, CrossChecks, CrossCheckUpdater, Turn, GameStepper*

Valors estudiats: Es col·loquen tota una sèrie de paraules al tauler

Efectes estudiats: Es pot veure com es van acumulant les fitxes al tauler correctament

4.2.9. placePieceThrowsExceptionIfNotConnectedToOtherWord

Objecte de prova: *PlaceActionMaker*

Altres elements integrats a la prova: *Board, Bag, DAWG, Player, WordGetter, PointCalculator, WordPlacer, PointCalculator, WordValidator, PiecesInHandGetter, MovementCleaner, PresentPiecesWordCompleter, CrossChecks, CrossCheckUpdater, Turn, GameStepper*

Valors estudiats: Es col·loquen dues paraules sense estar connectades entre elles

Efectes estudiats: Es llança l’excepció *WordNotConnectedToOtherWordsException*

4.2.10. placeWordWithBlankPiece

Objecte de prova: *PlaceActionMaker*

Altres elements integrats a la prova: *Board, Bag, DAWG, Player, WordGetter, PointCalculator, WordPlacer, PointCalculator, WordValidator, PiecesInHandGetter, MovementCleaner, PresentPiecesWordCompleter, CrossChecks, CrossCheckUpdater, Turn, GameStepper*

Valors estudiats: Es col·loca la paraula “HOLA”, on la “L” és un escarràs.

Efectes estudiats: Es col·loca la paraula damunt el tauler i se substitueix el caràcter de l'escarràs, per la lletra “L”.

4.2.11. placeFirstOutsideOfCenter

Objecte de prova: *PlaceActionMaker*

Altres elements integrats a la prova: *Board, Bag, DAWG, Player, WordGetter, PointCalculator, WordPlacer, PointCalculator, WordValidator, PiecesInHandGetter, MovementCleaner, PresentPiecesWordCompleter, CrossChecks, CrossCheckUpdater, Turn, GameStepper*

Valors estudiats: Es col·loca la paraula “HOLA” en horitzontal a la posició (1, 1).

Efectes estudiats: Es llança l'excepció *InitialMoveNotInCenterException*

4.3. TestSkipActionMaker

4.3.1. skipRunTriggersNextTurn

Objecte de prova: *SkipActionMaker*

Altres elements integrats a la prova: *Player, Endgame, Turn, IEndScreen, GameStepper, GamePlayerStub.*

Valors estudiats: El player1 ha saltat el seu torn.

Efectes estudiats: És el torn del player2 i no del player1.

4.4. TestAI

4.4.1. englishAIEmptyBoard

Objecte de prova: *EnglishAI*

Altres elements integrats a la prova: *DAWG, Board, WordAdder, Player, PointCalculator, Anchors, AnchorUpdater, WorldPlacer, WorldGetter, BoardViewStub, EnglishPiecesConverter, EnglishCrossChecks*

Valors estudiats: La *Board* comença buida. Els anchors tenen només la casella central inicial. Al DAWG hi ha afegida una paraula la qual és possible formar amb les peces afegides a la mà del bot.

Efectes estudiats: Funciona bé l'algorisme de la IA i s'aconsegueix trobar un moviment possible amb el *Board* buit.

4.4.2. catalanAIEmptyBoardNY

Objecte de prova: *CatalanAI*

Altres elements integrats a la prova: *DAWG, Board, WordAdder, Player, PointCalculator, Anchors, AnchorUpdater, WorldPlacer, WorldGetter, BoardViewStub, CatalanPiecesConverter, CatalanCrossChecks*

Valors estudiats: La *Board* comença buida. Els anchors tenen només la casella central inicial. Al DAWG hi ha afegida una paraula la qual és possible formar amb les peces afegides a la mà del bot. Aquesta paraula conté les lletres NY, caldrà la peça NY per formar-la.

Efectes estudiats: Funciona bé l'algorisme de la IA i s'aconsegueix trobar un moviment possible amb el *Board* buit amb la peça especial del català NY.

4.4.3. catalanAIEmptyBoardLGeminada

Objecte de prova: *CatalanAI*

Altres elements integrats a la prova: *DAWG, Board, WordAdder, Player, PointCalculator, Anchors, AnchorUpdater, WorldPlacer, WorldGetter, BoardViewStub, CatalanPiecesConverter, CatalanCrossChecks*

Valors estudiats: La *Board* comença buida. Els anchors tenen només la casella central inicial. Al DAWG hi ha afegida una paraula la qual és possible formar amb les peces afegides a la mà del bot. Aquesta paraula conté les lletres L·L, caldrà la peça L·L per formar-la.

Efectes estudiats: Funciona bé l'algorisme de la IA i s'aconsegueix trobar un moviment possible amb el *Board* buit amb la peça especial del català L·L.

4.4.4. catalanAIEmptyBoardNYendOfWord

Objecte de prova: *CatalanAI*

Altres elements integrats a la prova: *DAWG, Board, WordAdder, Player, PointCalculator, Anchors, AnchorUpdater, WorldPlacer, WorldGetter, BoardViewStub, CatalanPiecesConverter, CatalanCrossChecks*

Valors estudiats: La *Board* comença buida. Els anchors tenen només la casella central inicial. Al *DAWG* hi ha afegida una paraula la qual és possible formar amb les peces afegides a la mà del bot. Aquesta paraula conté les lletres NY al final, caldrà la peça NY per formar-la.

Efectes estudiats: Funciona bé l'algorisme de la IA i s'aconsegueix trobar un moviment possible amb el *Board* buit amb la peça especial del català NY que s'ha de col·locar a la posició final.

4.4.5. catalanAIEmptyBoardLGeminadaEndOfWord

Objecte de prova: *CatalanAI*

Altres elements integrats a la prova: *DAWG, Board, WordAdder, Player, PointCalculator, Anchors, AnchorUpdater, WorldPlacer, WorldGetter, BoardViewStub, CatalanPiecesConverter, CatalanCrossChecks*

Valors estudiats: La *Board* comença buida. Els anchors tenen només la casella central inicial. Al *DAWG* hi ha afegida una paraula la qual és possible formar amb les peces afegides a la mà del bot. Aquesta paraula conté les lletres L·L al final, caldrà la peça L·L per formar-la.

Efectes estudiats: Funciona bé l'algorisme de la IA i s'aconsegueix trobar un moviment possible amb el *Board* buit amb la peça especial del català L·L que s'ha de col·locar a la posició final.

4.4.6. spanishAIEmptyBoardRR

Objecte de prova: *SpanishAI*

Altres elements integrats a la prova: *DAWG, Board, WordAdder, Player, PointCalculator, Anchors, AnchorUpdater, WorldPlacer, WorldGetter, BoardViewStub, SpanishPiecesConverter, SpanishCrossChecks*

Valors estudiats: La *Board* comença buida. Els anchors tenen només la casella central inicial. Al *DAWG* hi ha afegida una paraula la qual és possible formar amb les peces afegides a la mà del bot. Aquesta paraula conté les lletres RR, caldrà la peça RR per formar-la.

Efectes estudiats: Funciona bé l'algorisme de la IA i s'aconsegueix trobar un moviment possible amb el *Board* buit amb la peça especial del castellà RR.

4.4.7. spanishAIEmptyBoardRRendOfWord

Objecte de prova: *SpanishAI*

Altres elements integrats a la prova: *DAWG, Board, WordAdder, Player, PointCalculator, Anchors, AnchorUpdater, WorldPlacer, WorldGetter, BoardViewStub, SpanishPiecesConverter, SpanishCrossChecks*

Valors estudiats: La *Board* comença buida. Els anchors tenen només la casella central inicial. Al *DAWG* hi ha afegida una paraula la qual és possible formar amb les peces afegides a la mà del bot. Aquesta paraula conté les lletres RR al final, caldrà la peça RR per formar-la.

Efectes estudiats: Funciona bé l'algorisme de la IA i s'aconsegueix trobar un moviment possible amb el *Board* buit amb la peça especial del castellà RR que s'ha de col·locar a la posició final.

4.4.8. spanishAIEmptyBoardLL

Objecte de prova: *SpanishAI*

Altres elements integrats a la prova: *DAWG, Board, WordAdder, Player, PointCalculator, Anchors, AnchorUpdater, WorldPlacer, WorldGetter, BoardViewStub, SpanishPiecesConverter, SpanishCrossChecks*

Valors estudiats: La *Board* comença buida. Els anchors tenen només la casella central inicial. Al *DAWG* hi ha afegida una paraula la qual és possible formar amb les peces afegides a la mà del bot. Aquesta paraula conté les lletres LL, caldrà la peça LL per formar-la.

Efectes estudiats: Funciona bé l'algorisme de la IA i s'aconsegueix trobar un moviment possible amb el *Board* buit amb la peça especial del castellà LL.

4.4.9. spanishAIEmptyBoardLLendOfWord

Objecte de prova: *SpanishAI*

Altres elements integrats a la prova: *DAWG, Board, WordAdder, Player, PointCalculator, Anchors, AnchorUpdater, WorldPlacer, WorldGetter, BoardViewStub, SpanishPiecesConverter, SpanishCrossChecks*

Valors estudiats: La *Board* comença buida. Els anchors tenen només la casella central inicial. Al *DAWG* hi ha afegida una paraula la qual és possible formar amb les peces afegides a la mà del bot. Aquesta paraula conté les lletres LL al final, caldrà la peça LL per formar-la.

Efectes estudiats: Funciona bé l'algorisme de la IA i s'aconsegueix trobar un moviment possible amb el *Board* buit amb la peça especial del castellà LL que s'ha de col·locar a la posició final.

4.4.10. spanishAIEmptyBoardCH

Objecte de prova: *SpanishAI*

Altres elements integrats a la prova: *DAWG, Board, WordAdder, Player, PointCalculator, Anchors, AnchorUpdater, WorldPlacer, WorldGetter, BoardViewStub, SpanishPiecesConverter, SpanishCrossChecks*

Valors estudiats: La *Board* comença buida. Els anchors tenen només la casella central inicial. Al *DAWG* hi ha afegida una paraula la qual és possible formar amb les peces afegides a la mà del bot. Aquesta paraula conté les lletres CH, caldrà la peça CH per formar-la.

Efectes estudiats: Funciona bé l'algorisme de la IA i s'aconsegueix trobar un moviment possible amb el *Board* buit amb la peça especial del castellà CH.

4.4.11. spanishAIEmptyBoardCHendOfWord

Objecte de prova: *SpanishAI*

Altres elements integrats a la prova: *DAWG, Board, WordAdder, Player, PointCalculator, Anchors, AnchorUpdater, WorldPlacer, WorldGetter, BoardViewStub, SpanishPiecesConverter, SpanishCrossChecks*

Valors estudiats: La *Board* comença buida. Els anchors tenen només la casella central inicial. Al *DAWG* hi ha afegida una paraula la qual és possible formar amb les peces afegides a la mà del bot. Aquesta paraula conté les lletres CH al final, caldrà la peça CH per formar-la.

Efectes estudiats: Funciona bé l'algorisme de la IA i s'aconsegueix trobar un moviment possible amb el *Board* buit amb la peça especial del castellà CH que s'ha de col·locar a la posició final.

4.4.12. englishAIHorizontal

Objecte de prova: *EnglishAI*

Altres elements integrats a la prova: *DAWG, Board, WordAdder, Player, PointCalculator, Anchors, AnchorUpdater, WorldPlacer, WorldGetter, BoardViewStub, EnglishPiecesConverter, EnglishCrossChecks*

Valors estudiats: La *Board* té una paraula en vertical. Els anchors i crosschecks estan actualitzats en funció d'aquesta paraula. Al *DAWG* hi ha afegida una paraula la qual és possible formar amb les peces afegides a la mà del bot en conjunt amb les col·locades al *Board*.

Efectes estudiats: Funciona bé l'algorisme de la IA i s'aconsegueix trobar un moviment possible aprofitant una paraula en joc.

4.4.13. **catalanAIHorizontalINYRETRUNSMAXMOVEPOINTS**

Objecte de prova: *CatalanAI*

Altres elements integrats a la prova: *DAWG, Board, WordAdder, Player, PointCalculator, Anchors, AnchorUpdater, WorldPlacer, WorldGetter, BoardViewStub, CatalanPiecesConverter, CatalanCrossChecks*

Valors estudiats: La *Board* té una paraula en vertical, aquesta conté NY. Els anchors i crosschecks estan actualitzats en funció d'aquesta paraula. Al DAWG hi ha afegides dues paraules amb NY, les quals són possibles formar amb les peces afegides a la mà del bot i aprofitant la peça NY ja al *Board*.

Efectes estudiats: Funciona bé l'algorisme de la IA i s'aconsegueix trobar un moviment possible aprofitant la peça NY de la paraula en joc. Dels 2 moviments possibles retorna el que aconsegueix més punts.

4.4.14. **catalanAIHorizontalLGeminada**

Objecte de prova: *CatalanAI*

Altres elements integrats a la prova: *DAWG, Board, WordAdder, Player, PointCalculator, Anchors, AnchorUpdater, WorldPlacer, WorldGetter, BoardViewStub, CatalanPiecesConverter, CatalanCrossChecks*

Valors estudiats: La *Board* té una paraula en vertical, aquesta conté L·L. Els anchors i crosschecks estan actualitzats en funció d'aquesta paraula. Al DAWG hi ha afegida una paraula amb L·L, la qual és possible formar amb les peces afegides a la mà del bot i aprofitant la peça L·L ja al *Board*.

Efectes estudiats: Funciona bé l'algorisme de la IA i s'aconsegueix trobar un moviment possible aprofitant la peça L·L de la paraula en joc.

4.4.15. **spanishAIHorizontalRR**

Objecte de prova: *SpanishAI*.

Altres elements integrats a la prova: *DAWG, Board, WordAdder, Player, PointCalculator, Anchors, AnchorUpdater, WorldPlacer, WorldGetter, BoardViewStub, SpanishPiecesConverter, SpanishCrossChecks*

Valors estudiats: La *Board* té una paraula en vertical, aquesta conté RR. Els anchors i crosschecks estan actualitzats en funció d'aquesta paraula. Al DAWG hi ha afegida una paraula amb RR, la qual és possible formar amb les peces afegides a la mà del bot i aprofitant la peça RR ja al *Board*.

Efectes estudiats: Funciona bé l'algorisme de la IA i s'aconsegueix trobar un moviment possible aprofitant la peça RR de la paraula en joc.

4.4.16. spanishAIHorizontalLL

Objecte de prova: *SpanishAI*

Altres elements integrats a la prova: *DAWG, Board, WordAdder, Player, PointCalculator, Anchors, AnchorUpdater, WorldPlacer, WorldGetter, BoardViewStub, SpanishPiecesConverter, SpanishCrossChecks*

Valors estudiats: La *Board* té una paraula en vertical, aquesta conté LL. Els anchors i crosschecks estan actualitzats en funció d'aquesta paraula. Al *DAWG* hi ha afegida una paraula amb LL, la qual és possible formar amb les peces afegides a la mà del bot i aprofitant la peça LL ja al *Board*.

Efectes estudiats: Funciona bé l'algorisme de la IA i s'aconsegueix trobar un moviment possible aprofitant la peça LL de la paraula en joc.

4.4.17. spanishAIHorizontalCH

Objecte de prova: *SpanishAI*

Altres elements integrats a la prova: *DAWG, Board, WordAdder, Player, PointCalculator, Anchors, AnchorUpdater, WorldPlacer, WorldGetter, BoardViewStub, SpanishPiecesConverter, SpanishCrossChecks*

Valors estudiats: La *Board* té una paraula en vertical, aquesta conté CH. Els anchors i crosschecks estan actualitzats en funció d'aquesta paraula. Al *DAWG* hi ha afegida una paraula amb CH, la qual és possible formar amb les peces afegides a la mà del bot i aprofitant la peça CH ja al *Board*.

Efectes estudiats: Funciona bé l'algorisme de la IA i s'aconsegueix trobar un moviment possible aprofitant la peça CH de la paraula en joc.

4.4.18. englishAIHorizontalwithBlank

Objecte de prova: *EnglishAI*.

Altres elements integrats a la prova: *DAWG, Board, WordAdder, Player, PointCalculator, Anchors, AnchorUpdater, WorldPlacer, WorldGetter, BoardViewStub, EnglishPiecesConverter, EnglishCrossChecks*

Valors estudiats: La *Noard* té una paraula en vertical. Els anchors i crosschecks estan actualitzats en funció d'aquesta paraula. Al *DAWG* hi ha afegida una paraula la qual és possible formar amb les peces afegides a la mà del bot en conjunt amb les col·locades al *Board*. El bot té a la mà 1 blank que serà necessari fer servir per completar el possible moviment.

Efectes estudiats: Funciona bé l'algorisme de la IA i s'aconsegueix trobar un moviment possible aprofitant una paraula en joc i fent servir un blank. El moviment retornat té al seu String word els caràcters originats del blank en minúscula.

4.4.19. catalanAIEmptyBoardNYwithBlank

Objecte de prova: *CatalanAI*

Altres elements integrats a la prova: *DAWG, Board, WordAdder, Player, PointCalculator, Anchors, AnchorUpdater, WorldPlacer, WorldGetter, BoardViewStub, SpanishPiecesConverter, SpanishCrossChecks*

Valors estudiats: La *Board* té una paraula en vertical. Els anchors i crosschecks estan actualitzats en funció d'aquesta paraula. Al *DAWG* hi ha afegida una paraula la qual és possible formar amb les peces afegides a la mà del bot en conjunt amb les col·locades al *Board*. El bot té a la mà 1 blank que serà necessari fer servir per completar el possible moviment i aquest haurà de ser usat com NY.

Efectes estudiats: Funciona bé l'algorisme de la IA i s'aconsegueix trobar un moviment possible aprofitant una paraula en joc i fent servir un blank com a NY. El moviment retornat té al seu String word els caràcters originats del blank en minúscula.

4.4.20. spanishAIEmptyBoardRRwithBlank

Objecte de prova: *SpanishAI*

Altres elements integrats a la prova: *DAWG, Board, WordAdder, Player, PointCalculator, Anchors, AnchorUpdater, WorldPlacer, WorldGetter, BoardViewStub, SpanishPiecesConverter, SpanishCrossChecks*

Valors estudiats: La *Board* té una paraula en vertical. Els anchors i crosschecks estan actualitzats en funció d'aquesta paraula. Al *DAWG* hi ha afegida una paraula la qual és possible formar amb les peces afegides a la mà del bot en conjunt amb les col·locades al *Board*. El bot té a la mà 1 blank que serà necessari fer servir per completar el possible moviment i aquest haurà de ser usat com RR.

Efectes estudiats: Funciona bé l'algorisme de la IA i s'aconsegueix trobar un moviment possible aprofitant una paraula en joc i fent servir un blank com a RR. El moviment retornat té al seu String word els caràcters originats del blank en minúscula.

4.4.21. englishAIVertical

Objecte de prova: *EnglishAI*

Altres elements integrats a la prova: *DAWG, Board, WordAdder, Player, PointCalculator, Anchors, AnchorUpdater, WorldPlacer, WorldGetter, BoardViewStub, EnglishPiecesConverter, EnglishCrossChecks*

Valors estudiats: La *Board* comença amb una paraula en vertical. Anchors i CrossChecks actualitzats en funció d'aquesta. Al *DAWG* hi ha afegida una paraula la qual és possible formar amb les peces afegides a la mà del bot i la paraula en vertical del *Board*.

Efectes estudiats: Funciona bé l'algorisme de la IA i s'aconsegueix trobar un moviment possible en vertical.

4.4.22. englishAladjacent

Objecte de prova: *EnglishAI*

Altres elements integrats a la prova: *DAWG, Board, WordAdder, Player, PointCalculator, Anchors, AnchorUpdater, WorldPlacer, WorldGetter, BoardViewStub, EnglishPiecesConverter, EnglishCrossChecks*

Valors estudiats: Paraules adjacents que no formen paraules vàlides de forma vertical, però la consecució de les seves lletres sí es valida..

Efectes estudiats: No adició de paraules on les lletres inserides són vàlides, però no formen una paraula vàlida.

4.4.23. dontRepeatWords

Objecte de prova: *EnglishAI*

Altres elements integrats a la prova: *DAWG, Board, WordAdder, Player, PointCalculator, Anchors, AnchorUpdater, WorldPlacer, WorldGetter, BoardViewStub, EnglishPiecesConverter, EnglishCrossChecks*

Valors estudiats: La *Board* comença amb la paraula VIOLIN en horitzontal. La IA no pot fer cap moviment donades les peces que té a la mà.

Efectes estudiats: L'algorisme no retorna la paraula VIOLIN i en canvi retorna null.

4.5. TestAnchors

4.5.1. initialAnchor

Objecte de prova: *AnchorUpdater*

Altres elements integrats a la prova: *Anchors, Board*

Efectes estudiats: La casella central del *Board* es troba com a anchor.

4.5.2. updateAnchorHor

Objecte de prova: *AnchorUpdater*

Altres elements integrats a la prova: *Anchors, Move*

Efectes estudiats: S'elimina l'anchor superposat al centre del *Board* i es creen els que emboliquen el moviment horitzontal.

4.5.3. updateAnchorVer

Objecte de prova: *AnchorUpdater*

Altres elements integrats a la prova: *Anchors, Move*

Efectes estudiats: S'elimina l'anchor superposat al centre del *Board* i es creen els que emboliquen el moviment vertical.

4.5.4. rotateAnchors

Objecte de prova: *AnchorUpdater*

Altres elements integrats a la prova: *Anchors, Move*

Efectes estudiats: Els anchors resultants d'un moviment en horitzontal després de la rotació han girat correctament.

4.6. TestCrossChecks

4.6.1. createCrossChecks

Objecte de prova: *EnglishCrossChecks*

Altres elements integrats a la prova: *Board*

Efectes estudiats: Es crea els CrossChecks amb mida i inicialització correctes.

4.6.2. updateCrossChecks

Objecte de prova: *CrossCheckUpdater*

Altres elements integrats a la prova: *Board, DAWG, WordAdder, EnglishCrossCheck*

Valors estudiats: El DAWG conté 4 paraules. 2 d'elles només es diferencien perquè a una li falta la lletra inicial i les altres 2 per l'absència de la lletra final. S'actualitzen els CrossChecks com si es realitzessin els moviments de les paraules curtes.

Efectes estudiats: La casella prèvia i posterior dels moviments realitzats tenen correctament actualitzats els CrossChecks en funció de les altres 2 paraules existents al diccionari.

4.6.3. rotateCrossChecks

Objecte de prova: *EnglishCrossChecks*

Altres elements integrats a la prova: *Board, DAWG, WordAdder, CrossCheckUpdater*

Efectes estudiats: En realitzar la rotació dels CrossChecks els valors resultants es troben a les posicions correctes.

4.7. TestBoard

4.7.1. standardBoardHasSize15

Objecte de prova: *StandardBoard*

Efectes estudiats: Es crea un tauler de mida 15

4.7.2. superBoardHasSize21

Objecte de prova: *SuperBoard*

Efectes estudiats: Es crea un tauler de mida 21

4.7.3. jrBoardHasSize11

Objecte de prova: *JuniorBoard*

Efectes estudiats: Es crea un tauler de mida 11

4.7.4. boardIsEmptyWhenNoPiecesPlaced

Objecte de prova: *Board*

Efectes estudiats: Per defecte un tauler és buit

4.7.5. boardIsNotEmptyWhenPiecesPlaced

Objecte de prova: *Board*

Valors estudiats: Es col·loca la fitxa "C" a la posició (0, 0) del tauler

Efectes estudiats: Quan hi ha una fitxa col·locada, el tauler deixa d'estar buit

4.7.6. placePieceAddsItToBoard

Objecte de prova: *Board*

Valors estudiats: Es col·loca la fitxa "C" a la posició (0, 0) del tauler

Efectes estudiats: Si es demana al tauler la fitxa present a la posició (0, 0) aquest retorna la fitxa "C"

4.7.7. placePiecePlacesCorrectPiece

Objecte de prova: *Board*

Valors estudiats: Es col·loca la fitxa "D" a la posició (0, 0) del tauler

Efectes estudiats: Si es demana al tauler la fitxa present a la posició (0, 0) aquest retorna la fitxa "D"

4.7.8. readEmptyCellFromBoardReturnsNull

Objecte de prova: *Board*

Efectes estudiats: Si es demana al tauler la fitxa present en una posició on no hi ha cap fitxa, aquest retorna *null*

4.7.9. centerTileIsNotCorrectWhenNotCenter

Objecte de prova: *Board*

Valors estudiats: Es busca si la casella (7, 0) és el centre del tauler

Efectes estudiats: El tauler retorna que aquella casella no és el centre

4.7.10. centerTileIsCorrectInStandardBoard

Objecte de prova: *StandardBoard*

Valors estudiats: Es busca si la casella (7, 7) és el centre del tauler

Efectes estudiats: El tauler retorna que aquella casella sí que és el centre

4.7.11. centerTileIsCorrectInSuperBoard

Objecte de prova: *SuperBoard*

Valors estudiats: Es busca si la casella (7, 7) és el centre del tauler

Efectes estudiats: El tauler retorna que aquella casella sí que és el centre

4.7.12. centerTileIsCorrectInJuniorBoard

Objecte de prova: *JuniorBoard*

Valors estudiats: Es busca si la casella (7, 7) és el centre del tauler

Efectes estudiats: El tauler retorna que aquella casella sí que és el centre

4.7.13. emptyCellReturnsTrueWhenCellsEmpty

Objecte de prova: *Board*

Efectes estudiats: Per defecte una casella qualsevol del tauler és buida

4.7.14. emptyCellReturnsFalseWhenCellsNotEmpty

Objecte de prova: *Board*

Valors estudiats: Es col·loca la fitxa "C" a la casella (1, 1)

Efectes estudiats: La casella (1, 1) deixa de ser detectada com a buida

4.7.15. isPremiumTileReturnsTrueWhenCellsPremium

Objecte de prova: *Board*

Efectes estudiats: La casella de la cantonada (0, 0) conté un bonificador

4.7.16. isPremiumTileReturnsFalseWhenCellsNotPremium

Objecte de prova: *Board*

Efectes estudiats: La casella (1, 0) no conté un bonificador

4.7.17. rotateBoardRotatesPiece

Objecte de prova: *Board*

Valors estudiats: Es col·loca la fitxa "A" a la casella (0, 0)

Efectes estudiats: Quan es rota el tauler, aquesta fitxa passa a estar a la casella (0, 14)

4.7.18. rotateBoardRotatesWord

Objecte de prova: *Board*

Valors estudiats: Es col·loca la paraula “HOLA” en vertical començant per la casella (4, 0)

Efectes estudiats: La paraula passa a estar escrita en horitzontal començant per la casella (0, 10)

4.8. TestPointCalculator

4.8.1. calculateStandardPointsWithNoPremium

Objecte de prova: *PointCalculator*

Altres elements integrats a la prova: *StandardBoard*, *WordGetter*

Valors estudiats: Paraula “TEST” on la suma de les seves puntuacions es 6.

Efectes estudiats: Càlcul correcte de la paraula de prova en caselles regulars.

4.8.2. calculatePointsWhenPlacingSinglePiece

Objecte de prova: *PointCalculator*

Altres elements integrats a la prova: *StandardBoard*, *WordGetter*

Valors estudiats: “AT” i “IT” ja presents al tauler, inserció de “B” per formar dues paraules.

Efectes estudiats: Càlcul correcte de la puntuació obtinguda.

4.8.3. calculatePointsWhenPlacingSinglePiece AlreadyOnBoard

Objecte de prova: *PointCalculator*

Altres elements integrats a la prova: *StandardBoard*, *WordGetter*

Valors estudiats: “BAT” i “BIT” ja presents al tauler, compartint la lletra “B”.

Efectes estudiats: Càlcul correcte de la puntuació obtinguda per una peça ja present.

4.8.4. calculateStandardPointsWithPremiumLetter

Objecte de prova: *PointCalculator*

Altres elements integrats a la prova: *StandardBoard*, *WordGetter*

Valors estudiats: Paraula “TEST” on la suma de les seves puntuacions es 6.

Efectes estudiats: Càlcul correcte de la paraula de prova en una casella de lletra premium.

4.8.5. calculateStandardPointsWithPremiumWord

Objecte de prova: *PointCalculator*

Altres elements integrats a la prova: *StandardBoard*, *WordGetter*

Valors estudiats: Paraula “TEST” on la suma de les seves puntuacions es 6.

Efectes estudiats: Càlcul correcte de la paraula de prova en una casella de paraula premium.

4.8.6. calculateBingoBonus

Objecte de prova: *PointCalculator*

Altres elements integrats a la prova: *StandardBoard*, *WordGetter*

Efectes estudiats: Càlcul correcte del bonus.

4.8.7. calculatePointsForTwoWordsSharingOnePiece

Objecte de prova: *PointCalculator*.

Altres elements integrats a la prova: *StandardBoard*, *WordGetter*.

Valors estudiats: Paraules “ONE” i “TEST”, 1 punt per peça.

Efectes estudiats: Càlcul correcte quan dues paraules comparteixen una peça ja present.

4.8.8. calculatePointsForTwoWordsSharing OnePieceNotPlaced

Objecte de prova: *PointCalculator*

Altres elements integrats a la prova: *StandardBoard*, *WordGetter*

Valors estudiats: Paraules “ONE” i “TEST”, 1 punt per peça.

Efectes estudiats: Càlcul correcte quan dues paraules comparteixen una peça no present.

4.8.9. calculatePointsForThreeWordsWith DifferentMultipliers

Objecte de prova: *PointCalculator*

Altres elements integrats a la prova: *StandardBoard*, *WordGetter*

Valors estudiats: Paraules “ONE”, “TEA”, “TITERED”, una peça de 2 punts, la resta 1.

Efectes estudiats: Càlcul correcte quan hi han 3 paraules amb diferents multiplicadors i ja hi son presents al tauler.

4.8.10. calculatePointsForThreeWordsWith DifferentMultipliersNotPlaced

Objecte de prova: *PointCalculator*

Altres elements integrats a la prova: *StandardBoard, WordGetter*

Valors estudiats: Paraules “ONE”, “TEA”, “TITERED”, una peça de 2 punts, la resta 1.

Efectes estudiats: Càlcul correcte quan hi han 3 paraules amb diferents multiplicadors i encara no totes son presents al tauler.

4.8.11. calculatePointsForThreeWordsWithGaps

Objecte de prova: *PointCalculator*

Altres elements integrats a la prova: *StandardBoard, WordGetter*

Valors estudiats: Paraules “ONE”, “TEA”, “TESTS”, 1 punt per peça.

Efectes estudiats: Càlcul correcte quan hi han 3 paraules diferents amb forats que ja hi son presents al tauler.

4.8.12. calculatePointsForThreeWordsWithGapsNotPlaced

Objecte de prova: *PointCalculator*

Altres elements integrats a la prova: *StandardBoard, WordGetter*

Valors estudiats: Paraules “ONE”, “TEA”, “TESTS”, 1 punt per peça.

Efectes estudiats: Càlcul correcte quan hi han 3 paraules diferents amb forats i encara no totes son presents al tauler.

4.8.13. calculatePointsForThreeWords

Objecte de prova: *PointCalculator*

Altres elements integrats a la prova: *StandardBoard, WordGetter*

Valors estudiats: Paraules “TEST”, “ANTS”, “ELEPHANT”, 1 punt per peça.

Efectes estudiats: Càlcul correcte quan es formen 3 paraules que ja hi son presents al tauler.

4.8.14. calculatePointsForThreeWordsNotPlaced

Objecte de prova: *PointCalculator*

Altres elements integrats a la prova: *StandardBoard*, *WordGetter*

Valors estudiats: Paraules “TEST”, “ANTS”, “ELEPHANT”, 1 punt per peça.

Efectes estudiats: Càlcul correcte quan es formen 3 paraules i encara no totes hi son presents al tauler.

4.9. TestPresentPiecesWordCompleter

4.9.1. checkWordCompleterReturnsExtendedWord

Objecte de prova: *PresentPiecesWordCompleter*

Altres elements integrats a la prova: *Board*, *WordGetter*

Valors estudiats: Adició de “S” a “TEST”.

Efectes estudiats: Correcte extensió de paraules ja presents.

4.9.2. checkWordCompleterReturnsCrossedWord

Objecte de prova: *PresentPiecesWordCompleter*

Altres elements integrats a la prova: *Board*, *WordGetter*

Valors estudiats: Adició de “EST” a “T” de forma creuada.

Efectes estudiats: Correcte extensió de paraules ja presents.

4.9.3. checkWordCompleterReturnsTwoCompletedWords WhenPlacingOnePiece

Objecte de prova: *PresentPiecesWordCompleter*.

Altres elements integrats a la prova: *Board*, *WordGetter*

Valors estudiats: Adició de “T” a dos “EST”.

Efectes estudiats: Correcte devolució de dues paraules.

4.9.4. **checkWordCompleterReturnsTwoCompletedWords WhenCompletingMorePieces**

Objecte de prova: *PresentPiecesWordCompleter*

Altres elements integrats a la prova: *Board, WordGetter*

Valors estudiats: Adició de “T” a dos “EST”.

Efectes estudiats: Correcte devolució de dues paraules.

4.9.5. **checkWordCompleterReturnsAllAdjacentPiecesOfTwoParallelWords**

Objecte de prova: *PresentPiecesWordCompleter*

Altres elements integrats a la prova: *Board, WordGetter*

Valors estudiats: “HOLA” i “CASA” adjacents.

Efectes estudiats: Correcte devolució de 5 paraules, la principal i les combinacions formades per les lletres adjacents.

4.10. **TestWordGetter**

4.10.1. **getVerticalWord**

Objecte de prova: *WordGetter*

Altres elements integrats a la prova: *Board*

Valors estudiats: Donat un tauler amb la paraula “TEST” en vertical, se li afegeix la lletra “S” al final mitjançant el WordGetter.

Efectes estudiats: El WordGetter retorna el conjunt de peces que formen la paraula completa “TESTS”.

4.10.2. **getVerticalWordWithTwoWordsInline**

Objecte de prova: *WordGetter*

Altres elements integrats a la prova: *Board*

Valors estudiats: Donat un tauler amb les paraules “NO” i “TEST” en vertical, se li afegeix la lletra “S” al final de TEST mitjançant el WordGetter.

Efectes estudiats: El WordGetter retorna el conjunt de peces que formen la paraula completa “TESTS”.

4.10.3. getHorizontalWord

Objecte de prova: *WordGetter*

Altres elements integrats a la prova: *Board*

Valors estudiats: Donat un tauler amb la paraula “TEST” en horitzontal, se li afegeix la lletra “S” al final mitjançant el WordGetter.

Efectes estudiats: El WordGetter retorna el conjunt de peces que formen la paraula completa “TESTS”.

4.10.4. getHorizontalWordWithTwoWordsInline

Objecte de prova: *WordGetter*

Altres elements integrats a la prova: *Board*

Valors estudiats: Donat un tauler amb les paraules “NO” i “TEST” en horitzontal, se li afegeix la lletra “S” al final de TEST mitjançant el WordGetter.

Efectes estudiats: El WordGetter retorna el conjunt de peces que formen la paraula completa “TESTS”.

4.10.5. getHorizontalWordThatWasAlreadyPlaced

Objecte de prova: *WordGetter*

Altres elements integrats a la prova: *Board*

Valors estudiats: Donat un tauler amb la paraula “NO” en horitzontal, se li intenta tornar a afegir la lletra “O” al final mitjançant el WordGetter.

Efectes estudiats: El WordGetter retorna el conjunt de peces que formen la paraula completa “NO”.

4.10.6. getHorizontalWordWithPreviousWordHavingSame Character

Objecte de prova: *WordGetter*

Altres elements integrats a la prova: *Board*

Valors estudiats: Donat un tauler amb la paraula “NO” i “TOAST” en horitzontal, se li intenta tornar a afegir la lletra “O” al final mitjançant el WordGetter. Mirar que el WordGetter no confongui la “O” a seleccionar.

Efectes estudiats: El WordGetter retorna el conjunt de peces que formen la paraula completa “TOAST”.

4.11. TestWordPlacer

4.11.1. placeWordVerticalPlacesWord

Objecte de prova: *WordPlacer*

Altres elements integrats a la prova: *Board, BoardViewStub, WordGetter, PointCalculator, Player*

Valors estudiats: "TEST" colocada de forma vertical amb inici a 0,0.

Efectes estudiats: Correcta addició al tauler.

4.11.2. placeWordHorizontalPlacesWord

Objecte de prova: *WordPlacer*

Altres elements integrats a la prova: *Board, BoardViewStub, WordGetter, PointCalculator, Player*

Valors estudiats: "TEST" colocada de forma horizontal amb inici a 0,0.

Efectes estudiats: Correcta addició al tauler.

4.11.3. noPlaceActionDoesntUpdateBoard

Objecte de prova: *WordPlacer*

Altres elements integrats a la prova: *Board, BoardViewStub, WordGetter, PointCalculator, Player*

Valors estudiats: No acció de posar.

Efectes estudiats: Correcte no actualització del tauler quan l'acció no implica posar res.

4.11.4. placeActionUpdatesBoard

Objecte de prova: *WordPlacer*

Altres elements integrats a la prova: *Board, BoardViewStub, WordGetter, PointCalculator, Player*

Valors estudiats: Acció de posar.

Efectes estudiats: Correcta actualització del tauler quan l'acció implica posar.

4.12. TestGamesPlayed

4.12.1. testRunReturnsCorrectGamesPlayedSorted

Objecte de prova: *GamesPlayedLeaderboard*

Altres elements integrats a la prova: *Leaderboard, Score & PlayerValuePair*

Valors estudiats: Diferents Score on l'únic contingut rellevant és el paràmetres playerName i la seva quantitat.

Efectes estudiats: Retorn d'una array ordenada amb els valors correctes, sense jugadors repetits i un reverse conseqüent.

4.13. TestGamesWon

4.13.1. testRunReturnsCorrectGamesWonSorted

Objecte de prova: *GamesWonLeaderboard*

Altres elements integrats a la prova: *Leaderboard, Score & PlayerValuePair*

Valors estudiats: Diferents Score on els continguts rellevants són els paràmetres isWinner, playerName i la seva quantitat.

Efectes estudiats: Retorn d'una array ordenada amb els valors correctes, sense jugadors repetits i un reverse conseqüent.

4.14. TestMaxScore

4.14.1. testRunReturnsCorrectMaxScoreSorted

Objecte de prova: *MaxScoreLeaderboard*

Altres elements integrats a la prova: *Leaderboard, Score & PlayerValuePair*

Valors estudiats: Diferents Score on els continguts rellevants són els paràmetres scoreValue, playerName i la seva quantitat.

Efectes estudiats: Retorn d'una array ordenada amb els valors correctes, sense jugadors repetits i un reverse conseqüent.

4.15. TestTotalScore

4.15.1. testRunReturnsCorrectTotalScoreSorted

Objecte de prova: *TotalScoreLeaderboard*.

Altres elements integrats a la prova: *Leaderboard*, *Score* & *PlayerValuePair*

Valors estudiats: Diferents *Score* on els continguts rellevants són els paràmetres *scoreValue*, *playerName* i la seva quantitat.

Efectes estudiats: Retorn d'una array ordenada amb els valors correctes, sense jugadors repetits i un reverse conseqüent.

4.16. TestWinRate

4.16.1. testRunReturnsCorrectWinRatesSorted

Objecte de prova: *WinRateLeaderboard*

Altres elements integrats a la prova: *Leaderboard*, *Score*, *PlayerValuePair* & *WinsGamesPair*

Valors estudiats: Diferents *Score* on els continguts rellevants són els paràmetres *isWinner*, *playerName* i la seva quantitat.

Efectes estudiats: Retorn d'una array ordenada amb els valors correctes, sense jugadors repetits i un reverse conseqüent.

4.17. TestDrawParser

4.17.1. testParseSingleLetter

Objecte de prova: *DrawParser*

Valors estudiats: Una lletra "A"

Efectes estudiats: Retorna un array de *Piece* contenenent únicament la fitxa "A"

4.17.2. testParseTwoLetters

Objecte de prova: *DrawParser*

Valors estudiats: Un string "A, B"

Efectes estudiats: Retorna un array de *Piece* contenenent les fitxes "A" i "B"

4.17.3. testParseSpecialDigraph

Objecte de prova: *DrawParser*

Valors estudiats: Un string "A, CH"

Efectes estudiats: Retorna un array de *Piece* contenenent les fitxes "A" i "CH"

4.17.4. testParseBlankTile

Objecte de prova: *DrawParser*

Valors estudiats: Un string "#, CH"

Efectes estudiats: Retorna un array de *Piece* contenenent les fitxes "#" de tipus escarràs i "CH"

4.18. TestMovementBoundsChecker

4.18.1. testExactJuniorHorizontalBounds

Objecte de prova: *MovementBoundsChecker*

Altres elements integrats a la prova: *Board, PiecesConverter, Movement*

Valors estudiats: Es col·loca a un tauler Junior la paraula "aaaaaaaaaaa" (11 lletres) en horitzontal començant a la posició (0, 0).

Efectes estudiats: Al tauler Junior hi ha la paraula "aaaaaaaaaaa" col·locada correctament.

4.18.2. testExactJuniorVerticalBounds

Objecte de prova: *MovementBoundsChecker*

Altres elements integrats a la prova: *Board, PiecesConverter, Movement*

Valors estudiats: Es col·loca a un tauler Junior la paraula "aaaaaaaaaaa" (11 lletres) en vertical començant a la posició (0, 0).

Efectes estudiats: Al tauler Junior hi ha la paraula "aaaaaaaaaaa" col·locada correctament.

4.18.3. testMoreThanJuniorHorizontalBounds

Objecte de prova: *MovementBoundsChecker*

Altres elements integrats a la prova: *Board, PiecesConverter, Movement*

Valors estudiats: Es col·loca a un tauler Junior la paraula "aaaaaaaaaaaaa" (12 lletres) en horitzontal començant a la posició (0, 0).

Efectes estudiats: Al tauler Junior no hi cap la paraula, no canvia.

4.18.4. testMoreThanJuniorVerticalBounds

Objecte de prova: *MovementBoundsChecker*

Altres elements integrats a la prova: *Board, PiecesConverter, Movement*

Valors estudiats: Es col·loca a un tauler Junior la paraula "aaaaaaaaaaaa" (12 lletres) en vertical començant a la posició (0, 0).

Efectes estudiats: Al tauler Junior no hi cap la paraula, no canvia.

4.18.5. testExactStandardHorizontalBounds

Objecte de prova: *MovementBoundsChecker*

Altres elements integrats a la prova: *Board, PiecesConverter, Movement*

Valors estudiats: Es col·loca a un tauler Standard la paraula "aaaaaaaaaaaaaaaa" (15 lletres) en horitzontal començant a la posició (0, 0).

Efectes estudiats: Al tauler Standard hi ha la paraula "aaaaaaaaaaaaaaaa" col·locada correctament.

4.18.6. testExactStandardVerticalBounds

Objecte de prova: *MovementBoundsChecker*

Altres elements integrats a la prova: *Board, PiecesConverter, Movement*

Valors estudiats: Es col·loca a un tauler Standard la paraula "aaaaaaaaaaaaaaaa" (15 lletres) en vertical començant a la posició (0, 0).

Efectes estudiats: Al tauler Standard hi ha la paraula "aaaaaaaaaaaaaaaa" col·locada correctament.

4.18.7. testMoreThanStandardHorizontalBounds

Objecte de prova: *MovementBoundsChecker*

Altres elements integrats a la prova: *Board, PiecesConverter, Movement*

Valors estudiats: Es col·loca a un tauler Standard la paraula "aaaaaaaaaaaaaaaa" (16 lletres) en horitzontal començant a la posició (0, 0).

Efectes estudiats: Al tauler Standard no hi cap la paraula, no canvia.

4.18.8. testMoreThanStandardVerticalBounds

Objecte de prova: *MovementBoundsChecker*

Altres elements integrats a la prova: *Board, PiecesConverter, Movement*

Valors estudiats: Es col·loca a un tauler Standard la paraula "aaaaaaaaaaaaaaaa" (16 lletres) en vertical començant a la posició (0, 0).

Efectes estudiats: Al tauler Standard no hi cap la paraula, no canvia.

4.18.9. testExactSuperHorizontalBounds

Objecte de prova: *MovementBoundsChecker*

Altres elements integrats a la prova: *Board, PiecesConverter, Movement*

Valors estudiats: Es col·loca a un tauler Super la paraula "aaaaaaaaaaaaaaaaaaaaaa" (21 lletres) en horitzontal començant a la posició (0, 0).

Efectes estudiats: Al tauler Super hi ha la paraula "aaaaaaaaaaaaaaaaaaaaaa" col·locada correctament.

4.18.10. testExactSuperVerticalBounds

Objecte de prova: *MovementBoundsChecker*

Altres elements integrats a la prova: *Board, PiecesConverter, Movement*

Valors estudiats: Es col·loca a un tauler Super la paraula "aaaaaaaaaaaaaaaaaaaaaa" (21 lletres) en vertical començant a la posició (0, 0).

Efectes estudiats: Al tauler Super hi ha la paraula "aaaaaaaaaaaaaaaaaaaaaa" col·locada correctament.

4.18.11. testMoreThanSuperHorizontalBounds

Objecte de prova: *MovementBoundsChecker*

Altres elements integrats a la prova: *Board, PiecesConverter, Movement*

Valors estudiats: Es col·loca a un tauler Super la paraula "aaaaaaaaaaaaaaaaaaaaaa" (22 lletres) en horitzontal començant a la posició (0, 0).

Efectes estudiats: Al tauler Super no hi cap la paraula, no canvia.

4.18.12. testMoreThanSuperVerticalBounds

Objecte de prova: *MovementBoundsChecker*

Altres elements integrats a la prova: *Board, PiecesConverter, Movement*

Valors estudiats: Es col·loca a un tauler Super la paraula "aaaaaaaaaaaaaaaaaaaaaa" (22 lletres) en vertical començant a la posició (0, 0).

Efectes estudiats: Al tauler Super no hi cap la paraula, no canvia.

4.19. TestMovementCleaner

4.19.1. test1PieceCleanerVertical

Objecte de prova: *MovementCleaner*

Altres elements integrats a la prova: *Piece, Pair, Board, PiecesConverter, Player, BoardViewStub, WordGetter, PointCalculator, WordPlacer, Movement*

Valors estudiats: Es col·loca la paraula "TEST" en direcció vertical començant a la posició (0,0). Al tauler hi ha col·locada ja una "T" a la casella (0, 0).

Efectes estudiats: Es retornen les peces "EST".

4.19.2. test2PiecesCleanerVertical

Objecte de prova: *MovementCleaner*

Altres elements integrats a la prova: *Piece, Pair, Board, PiecesConverter, Player, BoardViewStub, WordGetter, PointCalculator, WordPlacer, Movement*

Valors estudiats: Es col·loca la paraula "TEST" en direcció vertical començant a la posició (0,0). Al tauler hi ha col·locada ja la paraula "TE" començant a la casella (0, 0).

Efectes estudiats: Es retornen les peces "ST".

4.19.3. test3PiecesCleanerVertical

Objecte de prova: *MovementCleaner*

Altres elements integrats a la prova: *Piece, Pair, Board, PiecesConverter, Player, BoardViewStub, WordGetter, PointCalculator, WordPlacer, Movement*

Valors estudiats: Es col·loca la paraula "TEST" en direcció vertical començant a la posició (0,0). Al tauler hi ha col·locada ja la paraula "TES" començant a la casella (0, 0).

Efectes estudiats: Es retornen les peces "T".

4.19.4. testAllPiecesCleanerVertical

Objecte de prova: *MovementCleaner*

Altres elements integrats a la prova: *Piece, Pair, Board, PiecesConverter, Player, BoardViewStub, WordGetter, PointCalculator, WordPlacer, Movement*

Valors estudiats: Es col·loca la paraula "TEST" en direcció vertical començant a la posició (0,0). Al tauler hi ha col·locada ja la paraula "TEST" començant a la casella (0, 0).

Efectes estudiats: Es retorna un array buit de peces.

4.19.5. test1PiecesCleanerHorizontal

Objecte de prova: *MovementCleaner*

Altres elements integrats a la prova: *Piece, Pair, Board, PiecesConverter, Player, BoardViewStub, WordGetter, PointCalculator, WordPlacer, Movement*

Valors estudiats: Es col·loca la paraula "TEST" en direcció horitzontal començant a la posició (0,0). Al tauler hi ha col·locada ja una "T" a la casella (0, 0).

Efectes estudiats: Es retornen les peces "EST".

4.19.6. test2PiecesCleanerHorizontal

Objecte de prova: *MovementCleaner*

Altres elements integrats a la prova: *Piece, Pair, Board, PiecesConverter, Player, BoardViewStub, WordGetter, PointCalculator, WordPlacer, Movement*

Valors estudiats: Es col·loca la paraula "TEST" en direcció vertical començant a la posició (0,0). Al tauler hi ha col·locada ja la paraula "TE" començant a la casella (0, 0).

Efectes estudiats: Es retornen les peces "ST".

4.19.7. test3PiecesCleanerHorizontal

Objecte de prova: *MovementCleaner*

Altres elements integrats a la prova: *Piece, Pair, Board, PiecesConverter, Player, BoardViewStub, WordGetter, PointCalculator, WordPlacer, Movement*

Valors estudiats: Es col·loca la paraula "TEST" en direcció vertical començant a la posició (0,0). Al tauler hi ha col·locada ja la paraula "TES" començant a la casella (0, 0).

Efectes estudiats: Es retornen les peces "T".

4.19.8. testAllPiecesCleanerHorizontal

Objecte de prova: *MovementCleaner*

Altres elements integrats a la prova: *Piece, Pair, Board, PiecesConverter, Player, BoardViewStub, WordGetter, PointCalculator, WordPlacer, Movement*

Valors estudiats: Es col·loca la paraula “TEST” en direcció vertical començant a la posició (0,0). Al tauler hi ha col·locada ja la paraula “TEST” començant a la casella (0, 0).

Efectes estudiats: Es retorna un array buit de peces.

4.20. TestMovementParser

4.20.1. testMakeMoveReturnsCorrectWord

Objecte de prova: *MovementParser*

Valors estudiats: S'introdueix el moviment “TEST C1”

Efectes estudiats: S'exporta un *Movement* amb la paraula TEST

4.20.2. testMakeMoveReturnsCorrectPosition

Objecte de prova: *MovementParser*

Valors estudiats: S'introdueix el moviment “TEST C1”

Efectes estudiats: S'exporta un *Movement* amb la posició X=2 i Y=0

4.20.3. testMakeMoveReturnsCorrectPositionInverted

Objecte de prova: *MovementParser*

Valors estudiats: S'introdueix el moviment “TEST 1C”

Efectes estudiats: S'exporta un *Movement* amb la posició X=2 i Y=0

4.20.4. testMakeMoveReturnsCorrectDirectionHorizontal

Objecte de prova: *MovementParser*

Valors estudiats: S'introdueix el moviment “TEST C1”

Efectes estudiats: S'exporta un *Movement* amb direcció horitzontal

4.20.5. testMakeMoveReturnsCorrectDirectionVertical

Objecte de prova: *MovementParser*

Valors estudiats: S'introdueix el moviment “TEST 1C”

Efectes estudiats: S'exporta un *Movement* amb direcció vertical

4.21. TestPlayer

4.21.1. addScoreUpdatesPlayerScore

Objecte de prova: *Player*

Efectes estudiats: Donada una quantitat de Score a sumar s'actualitza correctament al player.

4.21.2. playerScoreEqualsZero

Objecte de prova: *Player*

Efectes estudiats: En crear-se el Player té Score a 0,

4.21.3. scoreAddsProperly

Objecte de prova: *Player*

Efectes estudiats: Després de vàries sumes Score s'acumula correctament.

4.22. TestBag

4.22.1. bagDefaultsToEmpty

Objecte de prova: *Bag*

Valors estudiats: Es crea una bag

Efectes estudiats: Es confirma que al crear una bag aquesta es buida

4.22.2. bagAddsOnePiece

Objecte de prova: *Bag*

Altres elements integrats a la prova: *Piece*

Valors estudiats: S'afegeix la lletra "a" a una bag buida.

Efectes estudiats: Es confirma que la bag conte únicament una instància de la lletra.

4.22.3. bagGetsOnePiece

Objecte de prova: *Bag*

Altres elements integrats a la prova: *Piece*

Valors estudiats: S'afegeix la lletra "a" a una bag buida i s'agafa una peça.

Efectes estudiats: Confirmem que obtenim la lletra "a".

4.22.4. **bagThrowsErrorIfTryGettingWhenEmpty**

Objecte de prova: *Bag*

Valors estudiats: Intentem agafar una peça d'una Bag buida creada prèviament.

Efectes estudiats: Confirmem que salta l'excepció *IllegalStateException*.

4.22.5. **bagThrowsErrorIfTryAddingNullPiece**

Objecte de prova: *Bag*

Valors estudiats: Intentem afegir una peça no acceptada (null) a una Bag creada prèviament.

Efectes estudiats: Confirmem que salta l'excepció *IllegalArgumentException*.

4.23. **TestBagFiller**

4.23.1. **bagsFilled**

Objecte de prova: *BagFiller*

Altres elements integrats a la prova: *PieceGenerator*, *Bag*

Valors estudiats: S'omple la bossa amb la informació de la peça "A 3 1".

Efectes estudiats: La bossa té 1 peça de lletra "A" i valor "3".

4.24. **TestDAWG**

4.24.1. **createDAWGproperly**

Objecte de prova: *DAWG*

Altres elements integrats a la prova: *WordValidator*

Efectes estudiats: El DAWG es crea correctament.

4.24.2. **addWord**

Objecte de prova: *WordAdder*

Altres elements integrats a la prova: *DAWG* i *WordValidator*

Efectes estudiats: *WordAdder* afegeix correctament una paraula donada i *WordValidator* la pot validar.

4.24.3. addMultipleWords

Objecte de prova: *WordAdder*

Altres elements integrats a la prova: *DAWG* i *WordValidator*

Efectes estudiats: *WordAdder* afegeix correctament diverses paraules donades i *WordValidator* les pot validar.

4.25. TestHandFiller

4.25.1. tryFillWithEmptyBagThrowsException

Objecte de prova: *HandFiller*

Altres elements integrats a la prova: *Bag*, *Player*

Valors estudiats: Es reparteixen 7 peces a un jugador “test” amb una bossa buida.

Efectes estudiats: Es llança l'excepció *AssertionError*.

4.25.2. tryFillWithNotEnoughPiecesInBag ThrowsException

Objecte de prova: *HandFiller*

Altres elements integrats a la prova: *Bag*, *Player*

Valors estudiats: Es reparteixen 7 peces a un jugador “test” amb insuficients peces a la bossa.

Efectes estudiats: Es llança l'excepció *AssertionError*.

4.25.3. fillHandFillsPlayerCorrectly

Objecte de prova: *HandFiller*

Altres elements integrats a la prova: *Bag*, *Player*

Valors estudiats: Es reparteixen 7 peces a un jugador “test” amb suficients peces a la bossa.

Efectes estudiats: El jugador té 7 peces a la seva mà.

4.25.4. fillHandFillsAllPlayersCorrectly

Objecte de prova: *HandFiller*

Altres elements integrats a la prova: *Bag, Player*

Valors estudiats: Es reparteixen 7 peces a player1 “test” i player2 “test” amb suficients peces a la bossa, 7 “A” i 7 “B”.

Efectes estudiats: El player1 té 7 peces a la seva mà i totes són “A”, el player2 té 7 peces a la seva mà i totes són “B”.

4.26. TestPiece

4.26.1. pieceCreator

Objecte de prova: *Piece*

Valors estudiats: Es crea una nova Piece “a” amb value 1.

Efectes estudiats: Confirmen que obtenim una Piece del caràcter “a” amb value 1.

4.26.2. pieceComparator

Objecte de prova: *Piece*

Valors estudiats: Es creen dos Piece’s “a” amb value 1 i “b” amb value 2.

Efectes estudiats: Es confirma que aquestes dues no són iguals.

4.26.3. pieceBlank

Objecte de prova: *Piece*

Valors estudiats: Es crea una Piece qualsevol amb el paràmetre isBlank com a True.

Efectes estudiats: Es confirma que el parametre isBlank és True per a aquesta Piece.

4.26.4. setLetterBlankPiece

Objecte de prova: *Piece*

Valors estudiats: Es crea una Piece que estigui en blanc i s’intenta modificar la lletra

Efectes estudiats: Es modifica la lletra de la peça en blanc per una “a”

4.27. TestPieceDrawer

4.27.1. swap1piece

Objecte de prova: *PieceDrawer*

Altres elements integrats a la prova: *Bag, Player, IRand*

Valors estudiats: S'intercanvia una peça "b" del player amb una nova peça de la bossa. A la bossa hi ha la peça "a".

Efectes estudiats: El player té a la seva mà la peça "a" i a la bossa hi ha la peça "b".

4.27.2. swap2pieces

Objecte de prova: *PieceDrawer*

Altres elements integrats a la prova: *Bag, Player, IRand*

Valors estudiats: S'intercanvia les peces "b" i "c" del player amb dues noves peces de la bossa. A la bossa hi ha les peces "a" i "d".

Efectes estudiats: El player té a la seva mà les peces "a" i "d" i a la bossa hi ha les peces "b" i "c".

4.27.3. throwExceptionIfNotEnoughPiecesInBag

Objecte de prova: *PieceDrawer*

Altres elements integrats a la prova: *Bag, Player, IRand*

Valors estudiats: S'intercanvia les peces "b" i "c" del player amb dues noves peces de la bossa. A la bossa hi ha la peça "a".

Efectes estudiats: Es llança l'excepció *NotEnoughPiecesInBagException*.

4.28. TestPieceGenerator

4.28.1. whenPassingDPieceStringGenerateDPiece

Objecte de prova: *PieceGenerator*

Altres elements integrats a la prova: *Pair*

Valors estudiats: Es crea una peça donada la informació de la peça "D 4 2".

Efectes estudiats: La peça amb lletra "D" s'ha creat.

4.28.2. **whenPassingBlankPieceStringGenerateBlankPiece**

Objecte de prova: *PieceGenerator*

Altres elements integrats a la prova: *Pair*

Valors estudiats: Es crea una peça donada la informació de la peça “# 2 0”.

Efectes estudiats: La peça amb lletra “#” s’ha creat. La peça és un comodí, per tant, *isBlank* és cert.

4.28.3. **whenPassingCPieceStringGenerateCPiece**

Objecte de prova: *PieceGenerator*

Altres elements integrats a la prova: *Pair*

Valors estudiats: Es crea una peça donada la informació de la peça “C 2 3”.

Efectes estudiats: La peça amb lletra “C” s’ha creat.

4.28.4. **whenPassingDPieceStringGenerateDValue**

Objecte de prova: *PieceGenerator*

Altres elements integrats a la prova: *Pair*

Valors estudiats: Es crea una peça donada la informació de la peça “D 4 2”.

Efectes estudiats: La peça amb valor 2 s’ha creat.

4.28.5. **whenPassingEPieceStringGenerateEValue**

Objecte de prova: *PieceGenerator*

Altres elements integrats a la prova: *Pair*

Valors estudiats: Es crea una peça donada la informació de la peça “E 12 1”.

Efectes estudiats: La peça amb valor 1 s’ha creat.

4.28.6. **whenPassingDPieceStringGenerateDCount**

Objecte de prova: *PieceGenerator*

Altres elements integrats a la prova: *Pair*

Valors estudiats: Es crea una peça donada la informació de la peça “D 4 2”.

Efectes estudiats: La quantitat de peces creades és 4.

4.28.7. **whenPassingEPieceStringGenerateECount**

Objecte de prova: *PieceGenerator*

Altres elements integrats a la prova: *Pair*

Valors estudiats: Es crea una peça donada la informació de la peça "E 12 1".

Efectes estudiats: La quantitat de peces creades és 12.

4.28.8. **parseMultiplePieces**

Objecte de prova: *PieceGenerator*

Altres elements integrats a la prova: *Pair*

Valors estudiats: Es creen peces donada la informació de les peces "E 12 1\nD 4 2".

Efectes estudiats: La peça amb lletra "E" s'ha creat. La peça amb lletra "D" s'ha creat.

4.29. **TestPiecesConverter**

4.29.1. **getLPieceFromLCharacter**

Objecte de prova: *PiecesConverter*

Altres elements integrats a la prova: *Piece*

Valors estudiats: Es converteix la paraula "L" en una peça.

Efectes estudiats: Es crea una peça amb lletra "L".

4.29.2. **getRPieceFromRCharacter**

Objecte de prova: *PiecesConverter*

Altres elements integrats a la prova: *Piece*

Valors estudiats: Es converteix la paraula "R" en una peça.

Efectes estudiats: Es crea una peça amb lletra "R".

4.29.3. **getPiecesFromWord**

Objecte de prova: *PiecesConverter*

Altres elements integrats a la prova: *Piece*

Valors estudiats: Es converteix la paraula "LA" en peces.

Efectes estudiats: Es crea una peça amb lletra "L". Es crea una peça amb lletra "A".

4.29.4. **getLGeminadaPiecesFromLGeminadaCharacterCAT**

Objecte de prova: *CatalanPiecesConverter*

Altres elements integrats a la prova: *Piece*

Valors estudiats: Es converteix la paraula “L·L” en una peça en la llengua del català.

Efectes estudiats: Es crea una peça amb lletra “L·L”.

4.29.5. **getNYPiecesFromNYCharacterCAT**

Objecte de prova: *CatalanPiecesConverter*

Altres elements integrats a la prova: *Piece*

Valors estudiats: Es converteix la paraula “NY” en una peça en la llengua del català.

Efectes estudiats: Es crea una peça amb lletra “NY”.

4.29.6. **getRRPiecesFromRRCharacterCAST**

Objecte de prova: *SpanishPiecesConverter*

Altres elements integrats a la prova: *Piece*

Valors estudiats: Es converteix la paraula “RR” en una peça en la llengua del castellà.

Efectes estudiats: Es crea una peça amb lletra “RR”.

4.29.7. **getLLPiecesFromLLCharacterCAST**

Objecte de prova: *SpanishPiecesConverter*

Altres elements integrats a la prova: *Piece*

Valors estudiats: Es converteix la paraula “LL” en una peça en la llengua del castellà.

Efectes estudiats: Es crea una peça amb lletra “LL”.

4.29.8. **getCHPiecesFromCHCharacterCAST**

Objecte de prova: *SpanishPiecesConverter*

Altres elements integrats a la prova: *Piece*

Valors estudiats: Es converteix la paraula “CH” en una peça en la llengua del castellà.

Efectes estudiats: Es crea una peça amb lletra “CH”.

4.29.9. **getPieceAsBlankPieceWhenCharacterIsLowercase**

Objecte de prova: *PiecesConverter*

Altres elements integrats a la prova: *Piece*

Valors estudiats: Es converteix la paraula “a” en una peça comodí.

Efectes estudiats: Es crea una peça (A, 0, true).

4.29.10. **getLgeminadaBlankPieceFromLgeminadaCharacterCAT**

Objecte de prova: *CatalanPiecesConverter*

Altres elements integrats a la prova: *Piece*

Valors estudiats: Es converteix la paraula “l·l” en una peça en la llengua del català.

Efectes estudiats: Es crea una peça (L·L, 0, true).

4.29.11. **getNYBlankPieceFromNYCharacterCAT**

Objecte de prova: *CatalanPiecesConverter*

Altres elements integrats a la prova: *Piece*

Valors estudiats: Es converteix la paraula “ny” en una peça en la llengua del català.

Efectes estudiats: Es crea una peça (NY, 0, true).

4.29.12. **getRRBlankPieceFromRRCharacterCAST**

Objecte de prova: *SpanishPiecesConverter*

Altres elements integrats a la prova: *Piece*

Valors estudiats: Es converteix la paraula “rr” en una peça en la llengua del castellà.

Efectes estudiats: Es crea una peça (RR, 0, true).

4.29.13. **getLLBlankPieceFromLLCharacterCAST**

Objecte de prova: *SpanishPiecesConverter*

Altres elements integrats a la prova: *Piece*

Valors estudiats: Es converteix la paraula “ll” en una peça en la llengua del castellà.

Efectes estudiats: Es crea una peça (LL, 0, true).

4.29.14. **getCHBlankPieceFromCHCharacterCAST**

Objecte de prova: *SpanishPiecesConverter*

Altres elements integrats a la prova: *Piece*

Valors estudiats: Es converteix la paraula “ch” en una peça en la llengua del castellà.

Efectes estudiats: Es crea una peça (CH, 0, true).

4.29.15. **getPieceWithDictionaryConvertsItWithProperScore**

Objecte de prova: *PiecesConverter*

Altres elements integrats a la prova: *Piece*

Valors estudiats: Es converteix la paraula “A” en una peça del diccionari. Al diccionari existeix la peça “A” de valor 5.

Efectes estudiats: Es crea una peça (A, 5, false).

4.29.16. **getSpecialCatalanPieceWithDictionaryConvertsItWithProperScore**

Objecte de prova: *CatalanPiecesConverter*

Altres elements integrats a la prova: *Piece*

Valors estudiats: Es converteix la paraula “NY” en una peça del diccionari en la llengua del català. Al diccionari existeix la peça “NY” de valor 5.

Efectes estudiats: Es crea una peça (NY, 5, false).

4.29.17. **getSpecialSpanishPieceWithDictionaryConvertsItWithProperScore**

Objecte de prova: *SpanishPiecesConverter*

Altres elements integrats a la prova: *Piece*

Valors estudiats: Es converteix la paraula “CH” en una peça del diccionari en la llengua del castellà. Al diccionari existeix la peça “CH” de valor 5.

Efectes estudiats: Es crea una peça (CH, 5, false).

4.29.18. `getSpecialSpanishBlankPieceWithDictionaryConver tsItWithNoScore`

Objecte de prova: *SpanishPiecesConverter*

Altres elements integrats a la prova: *Piece*

Valors estudiats: Es converteix la paraula “ch” en una peça del diccionari en la llengua del castellà. Al diccionari existeix la peça “CH” de valor 5.

Efectes estudiats: Es crea una peça amb lletra (CH, 5, true).

4.29.19. `getSpecialCatalanBlankPieceWithDictionaryConver tsItWithNoScore`

Objecte de prova: *CatalanPiecesConverter*

Altres elements integrats a la prova: *Piece*

Valors estudiats: Es converteix la paraula “ny” en una peça del diccionari en la llengua del català. Al diccionari existeix la peça “NY” de valor 5.

Efectes estudiats: Es crea una peça amb lletra (NY, 5, true).

4.30. `TestPiecesConverterFactory`

4.30.1. `generatePiecesConverterWithEnglishDictionary`

Objecte de prova: *PiecesConverterFactory*

Altres elements integrats a la prova: *IFileReader*, *PieceGenerator*, *PiecesConverter*

Valors estudiats: `Language.English`

Efectes estudiats: Es crea un `EnglishPiecesConverter`.

4.30.2. `generatePiecesConverterWithCatalanDictionary`

Objecte de prova: *PiecesConverterFactory*

Altres elements integrats a la prova: *IFileReader*, *PieceGenerator*, *PiecesConverter*

Valors estudiats: `Language.Catalan`

Efectes estudiats: Es crea un `CatalanPiecesConverter`.

4.30.3. generatePiecesConverterWithSpanishDictionary

Objecte de prova: *PiecesConverterFactory*

Altres elements integrats a la prova: *IFileReader, PieceGenerator, PiecesConverter*

Valors estudiats: *Language.Spanish*

Efectes estudiats: Es crea un *SpanishPiecesConverter*.

4.30.4. newPiecesConverterParsingWithCorrectLanguage Score

Objecte de prova: *PiecesConverterFactory*

Altres elements integrats a la prova: *IFileReader, PieceGenerator, PiecesConverter, Piece*

Valors estudiats: Donada la informació d'una peça "A 5 5", es converteix la paraula "A" en una peça.

Efectes estudiats: Es genera correctament una peça (A, 5) en català.

4.30.5. newPiecesConverterParsingWithMultipleScores

Objecte de prova: *PiecesConverterFactory*

Altres elements integrats a la prova: *IFileReader, PieceGenerator, PiecesConverter, Piece*

Valors estudiats: Donada la informació de peces "A 5 5\nB 3 2", es converteix la paraula "AB" en peces.

Efectes estudiats: Es genera correctament una peça (A, 5) i una altra (B, 2) en anglès.

4.31. TestPiecesInHandGetter

4.31.1. get1Piece

Objecte de prova: *PiecesInHandGetter*

Altres elements integrats a la prova: *Bag, Piece, Player, RandStrub*

Valors estudiats: S'agafa la peça (b, 1) de la mà del jugador. El jugador té 7 peces (b, 10).

Efectes estudiats: Es retorna 1 peça amb (b, 10).

4.31.2. **get1BlankPiece**

Objecte de prova: *PiecesInHandGetter*

Altres elements integrats a la prova: *Bag, Piece, Player, RandStrub*

Valors estudiats: S'agafa la peça amb lletra (b, 1, true) de la mà del jugador. El jugador té 6 peces (b, 10) i 1 peça (b, 0, true).

Efectes estudiats: Es retorna 1 peça amb lletra (b, 0, true).

4.31.3. **get1BlankPieceThrowsExceptionIfPlayerDoesNotHavePiece**

Objecte de prova: *PiecesInHandGetter*

Altres elements integrats a la prova: *Bag, Piece, Player, RandStrub*

Valors estudiats: S'agafa la peça (b, 1, true) de la mà del jugador. El jugador té 7 peces (b, 10).

Efectes estudiats: Es llança l'excepció *PlayerDoesNotHavePieceException*.

4.31.4. **rollbackWhenOnePieceNotFound**

Objecte de prova: *PiecesInHandGetter*

Altres elements integrats a la prova: *Bag, Piece, Player, RandStrub*

Valors estudiats: S'agafen 2 peces (a, 10) i (b, 10, true) de la mà del jugador. El jugador té una peça (a, 10) i (b, 10).

Efectes estudiats: Es llança l'excepció *PlayerDoesNotHavePieceException*. No s'ha tret cap peça de la mà del jugador.

4.31.5. **get1PieceRemovesOnePieceFromBag**

Objecte de prova: *PiecesInHandGetter*

Altres elements integrats a la prova: *Bag, Piece, Player, RandStrub*

Valors estudiats: S'agafa una peça (b, 1) de la mà del jugador. El jugador té a la mà 7 peces (b, 1). A la bossa de peces hi ha una peça (a, 1).

Efectes estudiats: La peça (a, 1) es treu de la bossa i s'afegeix a la mà del jugador.

4.31.6. **get2Pieces**

Objecte de prova: *PiecesInHandGetter*

Altres elements integrats a la prova: *Bag, Piece, Player, RandStrub*

Valors estudiats: S'agafen 2 peces (b, 1) de la mà del jugador. El jugador té a la mà 7 peces (b, 1). A la bossa hi ha 2 peces (a, 1).

Efectes estudiats: Es retornen dues peces (b, 1).

4.32. **TestPiecesInHandVerifier**

4.32.1. **getPieceABInHandAB**

Objecte de prova: *PiecesInHandVerifier*

Altres elements integrats a la prova: *Player, Piece, EnglishPiecesConverter*

Valors estudiats: Es verifica que les peces de la paraula "AB" estan a la mà del jugador. El jugador té a la mà les peces (A, 0) i (B, 0).

Efectes estudiats: Es retornen les peces (A, 0) i (B, 0).

4.32.2. **getPieceABInHandABCD**

Objecte de prova: *PiecesInHandVerifier*

Altres elements integrats a la prova: *Player, Piece, EnglishPiecesConverter*

Valors estudiats: Es verifica que les peces de la paraula "AB" estan a la mà del jugador. El jugador té a la mà les peces (A, 0) (B, 0) (C, 0) (D, 0).

Efectes estudiats: Es retornen les peces (A, 0) i (B, 0).

4.32.3. **getPieceNYInHandABNY**

Objecte de prova: *PiecesInHandVerifier*

Altres elements integrats a la prova: *Player, Piece, CatalanPiecesConverter*

Valors estudiats: Es verifica que la peça de la paraula "NY" està a la mà del jugador. El jugador té a la mà les peces (A, 0) (B, 0) (NY, 0).

Efectes estudiats: Es retornen la peça (NY, 0).

4.32.4. **getPieceCHInHandABCH**

Objecte de prova: *PiecesInHandVerifier*

Altres elements integrats a la prova: *Player, Piece, SpanishPiecesConverter*

Valors estudiats: Es verifica que la peça de la paraula “CH” està a la mà del jugador. El jugador té a la mà les peces (A, 0) (B, 0) (CH, 0).

Efectes estudiats: Es retornen la peça (CH, 0).

4.32.5. **getPieceLgeminadaInHandABLgeminada**

Objecte de prova: *PiecesInHandVerifier*

Altres elements integrats a la prova: *Player, Piece, CatalanPiecesConverter*

Valors estudiats: Es verifica que la peça de la paraula “L·L” està a la mà del jugador. El jugador té a la mà les peces (A, 0) (B, 0) (L·L, 0).

Efectes estudiats: Es retornen la peça (L·L, 0).

4.32.6. **getPieceRRInHandABRR**

Objecte de prova: *PiecesInHandVerifier*

Altres elements integrats a la prova: *Player, Piece, SpanishPiecesConverter*

Valors estudiats: Es verifica que la peça de la paraula “RR” està a la mà del jugador. El jugador té a la mà les peces (A, 0) (B, 0) (RR, 0).

Efectes estudiats: Es retornen la peça (RR, 0).

4.32.7. **getPieceNYCHRRInHandABNYCHRR**

Objecte de prova: *PiecesInHandVerifier*

Altres elements integrats a la prova: *Player, Piece, SpanishPiecesConverter*

Valors estudiats: Es verifica que la peça de la paraula “CHRR” està a la mà del jugador. El jugador té a la mà les peces (A, 0) (B, 0) (CH, 0) (RR, 0).

Efectes estudiats: Es retornen la peça (CH, 0). Es retornen la peça (RR 0).

4.33. TestSceneObject

4.33.1. createSceneObjectCallsConstructor

Objecte de prova: *SceneObjectStub*

Altres elements integrats a la prova: *Scene*

Valors estudiats: Es crida a *instantiate* (*SceneObjectStub*).

Efectes estudiats: El constructor de l'objecte s'ha invocat correctament.

4.33.2. createSceneObjectIsEnabledByDefault

Objecte de prova: *SceneObjectStub*

Altres elements integrats a la prova: *Scene*

Valors estudiats: Es crida a *instantiate* (*SceneObjectStub*).

Efectes estudiats: L'objecte es crea correctament i està en estat *enabled*.

4.33.3. disableObjectDisablesIt

Objecte de prova: *SceneObjectStub*

Altres elements integrats a la prova: *Scene*

Valors estudiats: Es crida a *instantiate* (*SceneObjectStub*). Es crida després a *disable()*

Efectes estudiats: L'objecte s'ha creat correctament i està en estat *disabled*.

4.33.4. sceneProcessCallsObjectProcessMethod

Objecte de prova: *SceneObjectStub*

Altres elements integrats a la prova: *Scene*, *SceneManager*

Valors estudiats: Es crida a *instantiate* (*SceneObjectStub*). Es configuren 2 unitats de temps i mentre està actiu crida *onProcess()*.

Efectes estudiats: Quan es crida el mètode *process* (*SceneManager*) el *process* de l'escena rep una crida.

4.33.5. **sceneProcessDoesNotCallObjectProcessMethodWhenDisabled**

Objecte de prova: *SceneObjectStub*

Altres elements integrats a la prova: *Scene, SceneManager*

Valors estudiats: S'instancia un objecte a l'escena i aquest es desactiva.

Efectes estudiats: En cridar el mètode process (SceneManager) el process de l'escena no rep cap crida.

4.33.6. **destroyObjectCallsDetachMethod**

Objecte de prova: *SceneObjectStub*

Altres elements integrats a la prova: *Scene*

Valors estudiats: Es crida a instantiate (SceneObjectStub). Es crida després a destroy()

Efectes estudiats: Es crida el mètode onDetach() de l'objecte.

4.34. **TestEndgame**

4.34.1. **TwoPlayersEndgameDueToSkipLimitReached**

Objecte de prova: *Endgame*

Altres elements integrats a la prova: *Player, Piece*

Valors estudiats: Es creen dos Players amb noms i Hands no buides i posem un skipCounter a 6

Efectes estudiats: Confirmem que obtenim True, ja que el skipCounter \geq numplayers*3

4.34.2. **TwoPlayersNoEndgameDueToSkipLimitReached**

Objecte de prova: *Endgame*

Altres elements integrats a la prova: *Player, Piece*

Valors estudiats: Es creen dos Players amb noms i Hands no buides i posem un skipCounter a 5

Efectes estudiats: Confirmem que obtenim False, ja que el skipCounter $<$ numplayers*3

4.34.3. TwoPlayersEndgameDueToSomeoneWithEmptyHand

Objecte de prova: *Endgame*

Altres elements integrats a la prova: *Player, Piece*

Valors estudiats: Es creen dos Players amb noms i Hands no buides menys un el qual té la seva hand buida

Efectes estudiats: Confirmem que la partida s'ha donat per acabada com que un player té la seva hand buida.

4.34.4. ThreePlayersEndgameDueToSkipLimitReached

Objecte de prova: *Endgame*

Altres elements integrats a la prova: *Player, Piece*

Valors estudiats: Es creen tres Players amb noms i Hands no buides i posem un skipCounter a 9

Efectes estudiats: Confirmem que obtenim True, ja que el skipCounter \geq numplayers*3

4.34.5. ThreePlayersNoEndgameDueToSkipLimitReached

Objecte de prova: *Endgame*

Altres elements integrats a la prova: *Player, Piece*

Valors estudiats: Es creen tres Players amb noms i Hands no buides i posem un skipCounter a 8

Efectes estudiats: Confirmem que obtenim False, ja que el skipCounter $<$ numplayers*3

4.34.6. ThreePlayersEndgameDueToSomeoneWithEmptyHand

Objecte de prova: *Endgame*

Altres elements integrats a la prova: *Player, Piece*

Valors estudiats: Es creen tres Players amb noms i Hands no buides menys un el qual té la seva hand buida

Efectes estudiats: Confirmem que la partida s'ha donat per acabada com que un player té la seva hand buida.

4.34.7. FourPlayersEndgameDueToSkipLimitReached

Objecte de prova: *Endgame*

Altres elements integrats a la prova: *Player* , *Piece*

Valors estudiats: Es creen quatre Players amb noms i Hands no buides i posem un skipCounter a 12

Efectes estudiats: Confirmem que obtenim True, ja que el skipCounter \geq numplayers*3

4.34.8. FourPlayersNoEndgameDueToSkipLimitReached

Objecte de prova: *Endgame*

Altres elements integrats a la prova: *Player*, *Piece*

Valors estudiats: Es creen quatre Players amb noms i Hands no buides i posem un skipCounter a 2

Efectes estudiats: Confirmem que obtenim False, ja que el skipCounter $<$ numplayers*3

4.34.9. FourPlayersEndgameDueToSomeoneWithEmptyHand

Objecte de prova: *Endgame*

Altres elements integrats a la prova: *Player*, *Piece*

Valors estudiats: Es creen quatre Players amb noms i Hands no buides menys un el qual té la seva hand buida

Efectes estudiats: Confirmem que la partida s'ha donat per acabada perquè un player té la seva hand buida.

5. Algorismes utilitzats

5.1. DAWG

Seguint el paper *The World's Fastest Scrabble Program*⁶ per emmagatzemar de manera eficient els diccionaris vam fer servir un *Directed Acyclic Word Graph*⁷ (DAWG). El principal avantatge que presenta aquesta estructura de dades respecte a un *Trie* és que redueix significativament l'ús de memòria mitjançant la compartició de nodes idèntics, evitant redundàncies.

Cada node del DAWG conté:

- **Caràcter:** El símbol que representa. Per representar peces que tinguin 2 o més caràcters (ex. l·l) caldrà més d'un node.
- **Successors:** HashMap<Character, Node>. Representa els nodes fills. Permet fer un accés ràpid al node successor de cert caràcter si el té.
- **Pare:** Referència al node pare.
- **Profunditat:** Distància des de l'arrel.
- **Hash:** Codi únic calculat en funció dels altres atributs del node (incloent-hi successors). El propòsit d'aquest codi hash és fer la comparació de si 2 nodes són iguals molt més ràpida, comparant només els seus hash en comptes de tots els seus valors.

El DAWG és, per tant:

- **Node arrel,** sense pare ni caràcter inicial
- **Nodes únics:** HashMap<Integer, Node> Per identificar i reutilitzar de manera eficient nodes iguals fent servir el seu hash.

Hi ha 2 maneres de mantenir poc espai quan creem un DAWG (i no fer un *Trie*), anar mirant cada vegada que afegim una nova paraula de no crear un node repetit, o després d'afegir tot el diccionari pronar els nodes repetits. Ens vam decantar per la primera opció.

Per afegir el nostre diccionari al DAWG ens aprofitem del fet que les paraules venen en ordre. És molt probable que 2 paraules consecutives comparteixin alguna part prefixa entre si, així que recorrem el DAWG fins a arribar a on difereix la nova respecte a l'anterior i comencem a afegir els nous nodes successors des d'allà.

En aquell moment creem un nou node amb els atributs necessaris que ens farien falta per afegir-lo als successors del node actual (i després continuem per aquest nou creat). Abans d'afegir-lo fem una consulta al HashMap de nodes únics amb el hash del node que hàgim

⁶ Appel, A. W., & Jacobson, G. J. (1988, 1 maig). The World's Fastest Scrabble Program.

⁷ Daciuk, J., Watson, B. W., Mihov, S., & Watson, R. E. (2000, 6 juliol). Incremental Construction of Minimal Acyclic Finite-State Automata.

creat per veure si hi havia algun amb el mateix hash, és a dir exactament igual. En cas que si el reutilitzem i no n'afegim un de nou.

En aquell moment com el codi hash dels nodes depèn dels seus successors i hem alterat els successors del node actual en afegir un de nou, així que actualitzem el seu codi hash i propaguem aquesta actualització recursivament al seu pare (fins a arribar a l'arrel).

Tot i que aquesta propagació i constant actualització dels codis hash sembla molt costosa té el principal *tradeoff* de després poder consultar molt eficientment al HashMap.

Una vegada hem arribat al final de la paraula a afegir al DAWG cal no oblidar-se de marcar l'últim node que sigui final de paraula.

La principal, tot i que no única, funcionalitat del DAWG és verificar si una paraula existeix o no. Per això començant des de l'arrel tractem de navegar tot el DAWG caràcter per caràcter fins a arribar al final. Si no aconseguim arribar al final o l'últim node no és un final de paraula la paraula no existeix.

5.2. Scrabble

Seguint el paper *The World's Fastest Scrabble Program*⁸ ara explicarem l'algorisme creat als anys 80 pel joc de taula Scrabble i com l'hem implementat.

Abans d'explicar el seu funcionament caldrà explicar les estructures de dades auxiliars que requereix aquest algorisme:

- **DAWG:** Explicat al punt 4.1.
- **Anchor:** Consisteix en un `ArrayList<Vector2>`. El primer i segon valor d'aquest `Vector 2` fan referència a posicions `x` i `y` del *Board*.

La principal funcionalitat aquesta classe és tenir emmagatzemades les posicions sobre les quals la IA pot començar a crear paraules, les caselles anchor. Inicialment, comença sent la casella central del *Board* (inicialitza en el seu `updater`). Cada vegada que es col·loca un moviment s'actualitza, eliminant les posicions sobre les quals s'ha efectuat el moviment i afegint les que l'envolten.

Tenir aquesta informació emmagatzemada i actualitzada en tot moment és molt més eficient que recalcul·lar-la iterant tot el *Board* cada vegada que la consultem.

- **CrossChecks:** Consisteix en una matriu de `BitSets` del size del *Board*. Cada bit representa una *piece*. La mida d'aquest bit set i la *piece* a la qual fan referència depèn de l'idioma seleccionat, per això tenim 3 tipus de crosschecks. Aquesta divisió per idiomes també és útil per mirar els casos de caràcters especials propis de cada idioma.

⁸ Appel, A. W., & Jacobson, G. J. (1988, 1 maig). The World's Fastest Scrabble Program.

El propòsit dels CrossChecks és evitar que la IA iteri sobre moviments que acabin deixant paraules invàlides al *Board* afegint peces al voltant de les ja formades. Al principi tots els BitSets comencen a 0. Cada vegada que es col·loca un moviment s'actualitzen els BitSets de les posicions afectades, posant a 1 els bits els quals fan referència a la peça que, col·locant-la a aquella posició invalidaria a alguna paraula ja formada al *Board*.

Els CrossChecks no s'actualitzen al voltant de cada paraula, a la casella d'abans de l'inici i després del final (si són vàlides). Les posicions paral·leles a la paraula col·locada, les files de dadalt i a baix (després veurem a la IA perquè 'només col·loquem paraules en horitzontal') ja es tracten a l'algorisme de la IA.

Una possible millora de la implementació perquè l'algorisme sigui una mica més ràpid seria fer que CrossChecks també tinguin emmagatzemada aquesta informació de les files paral·leles.

Tot i poder ser més extensa, igual que els anchors tenir aquesta informació emmagatzemada i actualitzada en tot moment és molt més eficient que recalcul·lar-la iterant tot el *Board* cada vegada que la volem consultar.

- **The World's Fastest Scrabble Program:** Una vegada explicades les estructures de dades utilitzades l'algorisme és de backtracking. El seu propòsit és donat l'estat del *Board* actual trobar el moviment que més punts aconsegueixi.

Els moviments que es poden efectuar essencialment són en horitzontal o vertical. La IA primerament farà una cerca de tots els moviments en horitzontal. Una vegada fet això aplicarem una rotació al *Board* i estructures de dades auxiliars, per fer una segona cerca, novament en horitzontal, tot i que ara realment gràcies a la rotació estarem considerant els moviments en vertical a l'orientació original.

Ens ha semblat més adient realitzar aquesta rotació que no modificar l'algorisme per tenir una còpia pràcticament idèntica, però que es realitzi en vertical. Considerem que els costos de realitzar les rotacions es negligible.

A aquesta segona passada emmagatzemem el millor moviment amb la seva component de direcció com a vertical. Si finalment el millor moviment s'efectua en vertical, aplicarem una transformació a les seves coordenades d'origen, per desfer que es van emmagatzemar com les horitzontals a l'estat amb rotació.

Tot i que no representi un gran impacte globalment, és millor que fer la rotació cada vegada que passi un moviment vertical a ser el que tingui millor puntuació, ja que només ho farem una vegada.

Les cerques amb backtracking consisteixen en, per cada casella anchor disponible tractar d'estendre cap a esquerra i dreta la possible paraula a formar fent servir paraules de la mà i existents al *Board*. Amb l'extensió cap a l'esquerra només es tindran en compte paraules a la *Board* o a la mà, però no les dues a la vegada.

Aquest algorisme és eficient, ja que, tracta d'evitar no començar a provar combinacions que no arribin a una paraula possible. Quan es realitza una extensió com es fa travessant el DAWG podem identificar ràpidament casos on no val la pena

continuar provant combinacions, ja que no hi ha cap manera de continuar avançant a través d'ell i, per tant, aconseguir una paraula vàlida. (també no provem a les cel·les amb CrossChecks les peces que tinguin el seu bit a 1).

Aquesta manera és molt més ràpida que, per exemple, fer totes les permutacions possibles de paraules que es poguessin formar i després comprovar-les una, ja que quasi totes no ens serviran per a res.

D'aquesta manera també podem anar comprovant si els nodes que travessem quan fem l'extensió cap a la dreta són finals de paraula per veure si és un moviment vàlid i anar actualitzant el millor moviment en cas que sí i aconseguir millor puntuació que l'actual.

Al dur a terme proves tot i els nostres tests vam tenir alguns problemes a solucionar. El primer va ser que es realitzaven moviments paral·lels a altres paraules ja col·locades que deixaven paraules invàlides. Com ja s'ha mencionat a CrossChecks ara comprovem manualment aquest fenomen.

Es podria millorar, ja que, ara per ara als idiomes amb peces especials es podria fer millor la comprovació d'aquestes paraules paral·leles, ja que, al cas concret de detectar que podrien causar descartem completament el moviment actual, perdent alguns possibles moviments.

L'altre va ser que la IA retornava moviments que eren tot paraules ja col·locades al *Board*. Una solució ràpida i poc costosa va ser guardar-se quantes peces tenia el jugador bot a l'inici del seu torn (no podia ser 7, ja que, als torns finals potser no hi ha prou peces i tenim menys de les habituals), i mirar al tractar de validar el moviment si havia baixat les actuals de la quantitat original.

6. Estructures de dades

6.1. Gestió d'escenes: *SceneManager*, *Scene* i *SceneObject*

Per gestionar l'estat del joc, hem organitzat les diferents parts del codi en escenes separades. En aquesta primera entrega, el joc disposa de dues escenes principals: l'escena de **menú** i l'escena de **joc**.

6.1.1. *SceneManager*

La classe *SceneManager* s'encarrega de controlar el bucle principal del programa, així com de gestionar el canvi d'escenes. Això inclou carregar noves escenes i descarregar les anteriors. Per fer-ho possible, *SceneManager* fa ús de dues classes fonamentals: *Scene* i *SceneObject*.

6.1.2. Scene

La classe *Scene* és una **classe abstracta** que proporciona la funcionalitat bàsica per gestionar els objectes d'una escena, com la seva **creació, actualització i destrucció**.

Cada nova escena ha d'heretar de *Scene*, i dins del seu constructor s'instancien tots els objectes que han d'estar presents mitjançant el mètode *instantiate()*, proporcionat per la mateixa classe base.

6.1.3. SceneObject

La classe *SceneObject* és també una **classe abstracta** que representa un objecte actiu dins d'una escena. Cada objecte pot implementar diversos callbacks per respondre als canvis d'estat i a l'evolució del joc. Els mètodes més rellevants són:

- **constructor()**: No pot tenir paràmetres. Es crida durant la instanciació de l'objecte.
- **onProcess(delta)**: S'executa en cada iteració del bucle principal. El paràmetre *delta* indica la diferència de temps respecte a l'última execució.
- **onEnable()**: S'executa quan l'objecte és activat. A partir d'aquest moment, rebrà actualitzacions a través de *onProcess(delta)*.
- **onDisable()**: Es crida quan l'objecte és desactivat.
- **onDetach()**: S'executa quan l'objecte és eliminat de l'escena.

A més, *SceneObject* ofereix mètodes com *instantiate()* i *destroy()* per facilitar la creació i eliminació d'objectes dins de l'escena.

6.2. Lectura de diccionaris i fitxes: DictionaryReader i PiecesReader

Per a la lectura dels diccionaris i peces hem decidit fer-ho en temps d'execució. Per gestionar aquesta funcionalitat de manera organitzada, s'han definit tres classes. Aquestes s'encarreguen de localitzar i llegir els fitxers corresponents segons l'idioma seleccionat.

6.2.1. LocaleReader

Aquesta és la classe base de la qual hereten les altres. El seu propòsit és proporcionar una funcionalitat genèrica per localitzar i llegir fitxers del sistema, situats dins d'un directori específic del projecte (*locales*).

Conté dos mètodes principals:

- **getAbsolutePath(String fileName):** Construeix la ruta absoluta cap a un fitxer donat, assumint que es troba dins de la jerarquia del paquet *edu/upc/prop/scrabble/locales*.
- **readFileToString(String filePath):** Llegeix i retorna el contingut del fitxer en format String amb codificació UTF-8. Si hi ha un error, retorna null.

Aquesta classe no s'instancia directament, sinó que serveix com a base perquè altres classes en puguin reutilitzar el comportament de lectura.

6.2.2. PiecesReader

Aquesta classe hereta de *LocaleReader* i s'encarrega de llegir els fitxers que contenen les peces (fitxes amb lletres i punts) que s'utilitzen en el joc.

Implementa un mètode públic *run(Language locale)* que selecciona el fitxer corresponent segons l'idioma:

- **Catalan** → *letrasCAT.txt*
- **Spanish** → *letrasCAST.txt*
- **English** → *letrasENG.txt*

Amb això, permet obtenir fàcilment les peces apropiades per a cada idioma, garantint que el joc pugui adaptar-se correctament a diferents localitzacions.

6.2.3. Pieces

També derivada de *LocaleReader*, aquesta classe està enfocada a llegir els fitxers de diccionaris, que contenen les paraules vàlides per a cada idioma del joc.

Té el mètode *run(Language locale)*, que retorna el contingut del fitxer corresponent:

- **Catalan** → *catalan.txt*
- **Spanish** → *castellano.txt*
- **English** → *english.txt*

6.3. Emmagatzematge i gestió de puntuacions: Score, Leaderboard i controladors

Per guardar tota la informació relacionada amb la puntuació i resultats de les partides i gestionar com mostrar-la disposem de dues classes a la capa de dades i diversos controladors a la capa de domini.

6.3.1. Score

Score és la classe més bàsica en aquest àmbit. Aquesta és un *public record* que guarda la puntuació obtinguda, si el jugador ha guanyat la partida i el seu nom.

6.3.2. Leaderboard

El següent esglaió en aquesta jerarquia és la classe leaderboard. Encarregada de guardar *Score* de diverses partides. Per aquest emmagatzematge hem utilitzat una *ArrayList*, que permet el creixement del nombre d'entrades gràcies a la seva gestió de memòria dinàmica.

6.3.3. Controladors

Per la gestió de com mostrar aquesta informació a l'usuari teníem diverses vies. Podíem emmagatzemar en diferents classes els *Score* amb els criteris adients. Això faria que cada vegada que es completa una partida s'hagués d'actualitzar cadascuna de les diferents classes. A més d'això molta de la informació emmagatzemada estaria duplicada. Com el volum de dades a processar mai serà molt gran vam optar per la via de crear controladors pels diferents mètodes de filtratge que processen les dades emmagatzemades a la *Leaderboard* i les retorni de la forma desitjada. D'aquesta manera podem guardar la informació relacionada a cada partida de forma senzilla i només operar quan es demana. A més aquesta metodologia facilita la inclusió de filtres extra o la introducció de variables extra a la classe *Score*, el que va molt agafats de la mà amb la filosofia amb la qual afrontem aquest projecte. Els mètodes de filtratge actualment són:

- **GamesPlayedLeaderboard:** retorna una array dels noms dels jugadors amb el seu nombre de partides jugades ordenada de forma descendent per aquesta quantitat.
- **GamesWinLeaderboard:** retorna una array dels noms dels jugadors amb el seu nombre de partides guanyades ordenada de forma descendent per aquesta quantitat.
- **MaxScoreLeaderboard:** retorna una array dels noms dels jugadors amb la seva màxima puntuació obtinguda ordenada de forma descendent per aquesta quantitat.
- **TotalScoreLeaderboard:** retorna una array dels noms dels jugadors amb la suma de les puntuacions obtingudes ordenada de forma descendent per aquesta quantitat.
- **WinRateLeaderboard:** retorna una array dels noms dels jugadors amb el seu percentatge de victòries ordenada de forma descendent per aquest percentatge.

Per ajudar l'execució d'aquest controlador disposem de dues classes auxiliars:

- **PlayerValuePair**: classe que guarda el nom del jugador i la variable a gestionar en format double. L'array que retornen els controladors és d'objectes d'aquesta classe.
- **GamesWinsPair**: classe que guarda el nombre de partides jugades i guanyades. Sempre està relacionada amb el nom d'un jugador i només s'utilitza per al controlador de percentatge de victòries.

6.4. Emmagatzematge dels moviments

Per emmagatzemar les dades més rellevants d'un moviment seguim el format estàndard abreujat del Scrabble⁹. Per a poder entendre millor aquest format, donem un parell d'exemples:

- **HOLa 2A**: En aquest cas, es col·locarà la paraula "HOLA" a la posició (2, A) en horitzontal. La lletra "A" d'aquest moviment és un escarràs.
- **ADEU F4**: En aquest cas, es col·locarà la paraula "ADEU" a la posició (F, 4) en vertical. No s'utilitza cap escarràs en aquest moviment.

Tal com es pot veure quan la primera coordenada del moviment és un número, aquest s'interpretarà com a moviment en horitzontal, en canvi, si la primera coordenada és una lletra, aquest s'interpretarà com un en vertical.

Els punts més importants d'aquest format són que els **escarrassos s'escriuen en minúscula** i que en la paraula del moviment **es representa la paraula sencera** i no les peces individuals a col·locar. En el moment d'efectuar el moviment, aquests dos factors es tenen en compte per seleccionar les peces que utilitza el jugador.

Internament, els moviments s'emmagatzemen en el record *Movement*. Aquest record conté la següent informació:

- **word**: String amb la paraula seguint el format explicat anteriorment.
- **x**: Enter amb la coordenada X de l'inici del moviment.
- **y**: Enter amb la coordenada Y de l'inici del moviment.
- **direction**: Direcció en la qual s'efectua el moviment (Horitzontal o Vertical)

Tal com es pot veure, aquí no utilitzem les coordenades amb el format abreujat del Scrabble, aquest format d'entrada únicament s'utilitza en la versió de la **vista per terminal**. Això és degut al fet que aquest format d'entrada de les coordenades únicament ens és útil en aquesta primera entrega.

⁹ [Scrabble/Rules](https://en.wikibooks.org/wiki/Scrabble/Rules#Example). (2024, 22 agost). Wikibooks.
<https://en.wikibooks.org/wiki/Scrabble/Rules#Example>

7. Relació de classes

Main - Gerard Gascón Moliné

7.1. Capa de dades

Anchor - Felipe Martínez Lassalle & Albert Usero Martínez

Movement - Gerard Gascón Moliné

Player - Albert Usero Martínez

7.1.1. Mòdul board

Board - Gerard Gascón Moliné

BoardType - Gerard Gascón Moliné

JuniorBoard - Gerard Gascón Moliné

PremiumTileType - Gerard Gascón Moliné

StandardBoard - Gerard Gascón Moliné

SuperBoard - Gerard Gascón Moliné

7.1.2. Mòdul crosschecks

CatalanCrossChecks - Felipe Martínez Lassalle & Albert Usero Martínez

CrossChecks - Felipe Martínez Lassalle & Albert Usero Martínez

EnglishCrossChecks - Felipe Martínez Lassalle & Albert Usero Martínez

SpanishCrossChecks - Felipe Martínez Lassalle & Albert Usero Martínez

7.1.3. Mòdul DAWG

DAWG - Albert Usero Martínez

Node - Albert Usero Martínez

7.1.4. Mòdul exceptions

PlayerDoesNotHavePieceException - Albert Usero Martínez

ScrabbleException - Gerard Gascón Moliné

7.1.5. Mòdul leaderboard

Leaderboard - Felipe Martínez Lassalle

Score - Felipe Martínez Lassalle

7.1.6. Mòdul pieces

Piece - Gina Escofet González

Bag - Gina Escofet González

7.1.7. Mòdul properties

GameProperties - Biel Pérez Silvestre

Language - Biel Pérez Silvestre

7.2. Capa de domini

7.2.1. Mòdul actionmaker

DrawActionMaker - Gina Escofet González

IHandDisplay - Gina Escofet González

PlaceActionMaker - Gerard Gascón Moliné

SkipActionMaker - Gerard Gascón Moliné

7.2.2. Mòdul AI

AI - Felipe Martínez Lassalle & Albert Usero Martínez

AnchorUpdater - Felipe Martínez Lassalle & Albert Usero Martínez

CatalanAI - Felipe Martínez Lassalle & Albert Usero Martínez

CrossCheckUpdater - Felipe Martínez Lassalle & Albert Usero Martínez

EnglishAI - Felipe Martínez Lassalle & Albert Usero Martínez

SpanishAI - Felipe Martínez Lassalle & Albert Usero Martínez

7.2.3. Mòdul board

IBoard - Gerard Gascón Moliné

PointCalculator - Gerard Gascón Moliné

PresentPiecesWordCompleter - Gerard Gascón Moliné

WordGetter - Gerard Gascón Moliné

WordPlacer - Gerard Gascón Moliné

7.2.4. Mòdul DAWG

WordAdder - Albert Usero Martínez

WordValidator - Albert Usero Martínez

7.2.5. Mòdul exceptions

InitialMoveNotInCenterException - Gerard Gascón Moliné

MovementOutsideOfBoardException - Gerard Gascón Moliné

NotEnoughPiecesInBagException - Gerard Gascón Moliné

WordDoesNotExistException - Gerard Gascón Moliné

WordNotConnectedToOtherWordsException - Gerard Gascón Moliné

7.2.6. Mòdul game

GameStepper - Biel Pérez Silvestre

IEndScreen - Biel Pérez Silvestre

7.2.7. Mòdul leaderboard

GamesPlayedLeaderboard - Felipe Martínez Lassalle

GamesWinPair - Felipe Martínez Lassalle

GamesWonLeaderboard - Felipe Martínez Lassalle

MaxScoreLeaderboard - Felipe Martínez Lassalle

PlayerValuePair - Felipe Martínez Lassalle

TotalScoreLeaderboard - Felipe Martínez Lassalle

WinRateLeaderboard - Felipe Martínez Lassalle

7.2.8. Mòdul movement

MovementBoundsChecker - Gina Escofet González

MovementCleaner - Gina Escofet González

7.2.9. Mòdul pieces

BagFiller - Gina Escofet González

CatalanPiecesConverter - Gina Escofet González

EnglishPiecesConverter - Gina Escofet González

HandFiller - Gerard Gascón Moliné

IFileReader - Gerard Gascón Moliné

PieceDrawer - Gina Escofet González

PieceGenerator - Gerard Gascón Moliné

PiecesConverter - Gina Escofet González

PiecesConverterFactory - Gerard Gascón Moliné

PiecesInHandGetter - Gina Escofet González

PiecesInHandVerifier - Gina Escofet González

SpanishPiecesConverter - Gina Escofet González

7.2.10. Mòdul turns

Endgame - Biel Pérez Silvestre

IGamePlayer - Gerard Gascón Moliné

Turn - Biel Pérez Silvestre

TurnResult - Gerard Gascón Moliné

7.3. Capa de presentació

7.3.1. Mòdul localization

DictionaryReader - Gerard Gascón Moliné

LocaleReader - Gerard Gascón Moliné

PiecesReader - Gerard Gascón Moliné

7.3.2. Mòdul scenes

Scene - Gerard Gascón Moliné

SceneManager - Gerard Gascón Moliné

SceneObject - Gerard Gascón Moliné

7.3.2.1. Mòdul excepcions

SceneObjectWithParametrizedConstructorException - Gerard Gascón Moliné

7.3.3. Mòdul terminal

BoardView - Gerard Gascón Moliné

EndScreen - Gina Escofet González

GameCreator - Biel Pérez Silvestre

LanguageSetter - Biel Pérez Silvestre

Player Setter - Biel Pérez Silvestre

SizeSetter - Biel Pérez Silvestre

TileConverter - Gerard Gascón Moliné

7.3.3.1. Mòdul actionmaker

HandView - Gina Escofet González

7.3.3.2. Mòdul movements

DrawParser - Gerard Gascón Moliné

MovementParser - Gerard Gascón Moliné

7.3.3.3. Mòdul players

AIPlayerObject - Felipe Martínez Lassalle

HumanPlayerObject - Gerard Gascón Moliné

PlayerObject - Gerard Gascón Moliné

7.3.3.4. Mòdul scenes

GameScene - Gerard Gascón Moliné

MenuScene - Gerard Gascón Moliné

7.3.3.5. Mòdul utils

Reader - Gerard Gascón Moliné

7.4. Testing

7.4.1. Mòdul actionmaker

TestDrawActionMaker - Gina Escofet González
TestPlaceActionMaker - Gerard Gascón Moliné
TestSkipActionMaker - Gerard Gascón Moliné

7.4.2. Mòdul AI

TestAI - Felipe Martínez Lassalle & Albert Usero Martínez
TestAnchors - Felipe Martínez Lassalle & Albert Usero Martínez
TestCrossChecks - Felipe Martínez Lassalle & Albert Usero Martínez

7.4.3. Mòdul board

TestBoard - Gerard Gascón Moliné
TestPointCalculator - Gerard Gascón Moliné
TestPresentPiecesWordCompleter - Gerard Gascón Moliné
TestWordGetter - Gerard Gascón Moliné
TestWordPlacer - Gerard Gascón Moliné

7.4.4. Mòdul leaderboard

TestGamesPlayed - Felipe Martínez Lassalle
TestGamesWon - Felipe Martínez Lassalle
TestMaxScore - Felipe Martínez Lassalle
TestTotalScore - Felipe Martínez Lassalle
TestWinRate - Felipe Martínez Lassalle

7.4.5. Mòdul movement

TestDrawParser - Gerard Gascón Moliné
TestMovementBoundsChecker - Gina Escofet González
TestMovementCleaner - Gina Escofet González
TestMovementParser - Gerard Gascón Moliné
TestPlayer - Albert Usero Martínez

7.4.6. Mòdul pieces

TestBag - Gina Escofet González
TestBagFiller - Gina Escofet González
TestDAWG - Albert Usero Martínez
TestHandFiller - Gerard Gascón Moliné
TestPiece - Gina Escofet González
TestPieceDrawer - Gina Escofet González
TestPieceGenerator - Gerard Gascón Moliné
TestPiecesConverter - Gina Escofet González
TestPiecesConverterFactory - Gerard Gascón Moliné

TestPiecesInHandGetter - Gina Escofet González
TestPiecesInHandVerifier - Gina Escofet González

7.4.7. Mòdul scenes

TestSceneObject - Gerard Gascón Moliné

7.4.8. Mòdul stubs

BoardViewStub - Gerard Gascón Moliné
EndScreenStub - Gina Escofet González
GamePlayerStub - Gerard Gascón Moliné
PiecesReaderStub - Gerard Gascón Moliné
RandStub - Gerard Gascón Moliné
SceneObjectStub - Gerard Gascón Moliné
SceneStub - Gerard Gascón Moliné

7.4.9. Mòdul turns

TestEndgame - Biel Pérez Silvestre

7.5. Paquet d'utilitats

Direction - Gerard Gascón Moliné
IRand - Gerard Gascón Moliné
Pair - Gerard Gascón Moliné
Rand - Gerard Gascón Moliné
Vector2 - Gerard Gascón Moliné